

Andrea Bianchini

DIDATTICA E PROGRAMMAZIONE

Edizione 1, 19 ottobre 2017

“Non è stato ancora dimostrato che l'intelligenza abbia qualche valore nella sopravvivenza.”

ARTHUR CHARLES CLARKE

INDICE

Prefazione Pag. 5

Capitolo 1 – La didattica

1.1 – Cosa ci spinge alla ricerca del sapere ?	Pag. 6
1.2 – Efficienza della conoscenza.	Pag. 8
1.3 – Penso dunque sono.	Pag.11
1.4 – I simboli.	Pag.13
1.5 – Patologia dell'insegnamento.	Pag.15
1.6 – Cosa farai da grande?	Pag.16
1.7 – Psicologia della didattica.	Pag.18
1.8 – Mi faccia un esempio.	Pag.20
1.9 – Ambasciator non porta pene.	Pag.22

Capitolo 2 - Esempi

2.1 – Proviamo ad insegnare.	Pag.23
2.2 – Informazione.	Pag.24
2.3 – Informatica	Pag.26
2.4 – Computer.	Pag.28

Capitolo 3 – La programmazione

3.1 – Un po' di storia.	Pag.29
3.2 – La mia storia di programmazione.	Pag.31
3.3 – Che cosa è un programma?	Pag.33

3.4 – I linguaggi di programmazione.	Pag.35
3.5 – Dati e funzioni.	Pag.39
3.6 – La programmazione ad oggetti.	Pag.41
3.7 – Gli algoritmi.	Pag.43
3.8 – Differenza tra un App ed una WebApp.	Pag.45
3.9 – Big data.	Pag.47
3.10 – La teoria della complessità.	Pag.48
3.11 – Colleghiamo un po le cose.	Pag.49

Capitolo 4 – Esempi

4.1 – Esempi di semplici programmi	Pag.51
4.2 – BASIC	Pag.52
4.3 – PASCAL	Pag.53
4.4 – C++	Pag.54
4.5 – ASP	Pag.55
4.6 – PHP	Pag.56
4.7 – JSP	Pag.57
4.8 – VB.NET	Pag.58
4.9 – C#	Pag.59
4.10 – ASP.NET	Pag.60
4.11 - Algoritmo di ordinamento (Bubble Sort).	Pag.61

Prefazione

Questo libro raccoglie in un'unica edizione i miei due libri 'la didattica' e 'la programmazione' più integrazioni realizzate a seguito del feedback dei miei lettori. L'idea di unire i miei due precedenti libri in un unico libro non era a me nuova, ma l'impulso determinante me lo ha fornito il Dott. Claudio Santilli, medico psichiatra presso il dipartimento di salute mentale di muraglia a Pesaro. Inoltre, dopo aver consegnato alcune copie dei miei libri 'la didattica' e 'la programmazione' ad amici e conoscenti, ho raccolto le loro opinioni, dubbi e perplessità, che mi hanno consentito di integrare con ulteriori scritti alcuni punti che non risultavano chiari se non addirittura non trattati. In particolare vorrei ringraziare il mio caro amico Franco Marino che è per me sempre fonte di ispirazione. Un ringraziamento al sito [ILMIOLIBRO](#), che mi è stato fatto conoscere dal Dott. Santilli, sul quale sono stati recensiti e commentati i miei libri consentendomi di avere delle opinioni con cui migliorarmi, o, almeno, provarci. Ringrazio anche mio nipote, l'ing. Davide Bianchini, che ha preso a cuore la mia attività di scrittore cercando di coinvolgermi in una attività di distribuzione dei miei libri. Infine ringrazio i miei genitori Marcello e Giuliana. Sembra incredibile ma la cosa più difficile in questo libro è stata trovare un titolo adatto.

Andrea Bianchini

Capitolo 1 – La didattica.

1.1 - Cosa ci spinge alla ricerca del sapere ?

Sicuramente, se pensiamo di accingerci ad una attività di apprendimento è perché dentro di noi qualcosa ci spinge a farlo. Non sto parlando del pezzo di carta che ci verrà rilasciato al termine di un percorso di apprendimento ma di qualcosa di più intimo, innato, in misura diversa in ciascuno di noi. Il sapere, demonizzato da taluni, è il comune divisore della specie umana. Persino i cani si chiedono perché arriva la notte, il buio, e poi, come per magia, risorge il sole. Tutti i giorni, dall'inizio del tempo sulla terra.

Bisogna cercare le origini di questa fame ancestrale di conoscenza nell'evoluzione della crescita dell'individuo sin dal grembo materno. Ma forse questo istinto ci deriva ereditariamente dal nostro DNA, come risultato di un meccanismo di adattamento mirato alla sopravvivenza. Se devo combattere con un animale feroce ne devo studiare e conoscere il comportamento, punti di forza e debolezze per schivare gli uni ed aggredire le seconde. Se devo proteggermi dalle variazioni climatiche ne devo prevedere l'evoluzione in modo da farmi trovare preparato. Tutto ciò ha un comune denominatore: la necessità di conoscere per riuscire ad adattarci all'ambiente senza soccombere. La conoscenza è il nostro sistema immunitario contro le aggressioni della vita. E si sa, di generazione in generazione i comportamenti più frequenti e di successo degli individui di una popolazione dopo mutazioni e ricombinazioni dovute all' ambiente ma

in parte anche alla casualità vengono somatizzati dal DNA di quella popolazione. Ora, è chiaro che non si tratta di somatizzazione bensì di selezione; sopravvivono gli individui che riescono a tramandare, quindi il risultato indiretto è la somatizzazione da parte del sistema del comportamento. Per successo di un individuo non intendo la capacità di far carriera in un contesto sociale piuttosto che un altro o la capacità di accumulare denaro ma la capacità di far valere le proprie tendenze ed aspirazioni sopravvivendo e tramandando l'utilità delle proprie esperienze a beneficio della collettività, perché questo significa sopravvivere: tramandare le proprie esperienze, non imponendole e non per creare cloni di noi stessi bensì 'motori' che possano riuscire a funzionare con qualsiasi carburante ed in qualsiasi ambiente riuscendo ad apportare un miglioramento al nostro DNA.

1.2 – Efficienza della conoscenza.

Una delle critiche più comuni mosse da taluni all'acquisizione di conoscenza è l'inutilità dello studiare oltre certi limiti o studiare materie che poco abbiano a che fare con la materia, insomma, materie che non servano a fare lavorare le mani; costruire case, palazzi, ponti piuttosto che ferrovie. Cito alcune frasi; “Quello ha studiato troppo”, “Quello è troppo intelligente”, “Tutto teoria, niente pratica...”, “Bisogna studiare il giusto!!!”, “Ma va a lavorare! “

Quindi a questo punto risulta utile introdurre il concetto di “efficienza della conoscenza”, e cioè; fino a che punto è utile acquisire nuova conoscenza? Esiste un limite alla quantità di conoscenza che un individuo possa acquisire? Ma per usare una definizione più scientifica; qual è il ritorno dell'investimento in conoscenza?

Certo, vi è un limite fisico alla quantità di conoscenza che un singolo individuo può gestire. Se parliamo di conoscenza nozionistica vi è un limite al numero di nozioni che un singolo individuo è in grado di memorizzare, per non parlare del fatto che le nozioni si deteriorano e necessitano di essere rinfrescate, se invece parliamo di conoscenza matematica vi è un limite al grado di astrazione del ragionamento che un singolo individuo è in grado di raggiungere. Se l'istinto di conoscenza è scritto sul nostro DNA, è possibile ipotizzare una degenerazione, come per tutti gli altri elementi della sfera “uomo”, del meccanismo di apprendimento di nuova conoscenza? Cioè, è mai possibile che un eccesso di conoscenza o una ‘cattiva’

conoscenza possano in qualche modo danneggiare la nostra psiche? Per intenderci; la conoscenza è come il cibo? Assimilarne in eccesso o di cattiva qualità, può danneggiare i nostri processi mentali?

Io penso che il primo ritorno in investimento nell'acquisizione di nuova conoscenza sia un aumento del proprio welfare. Secondariamente, nuova conoscenza significa nuove armi per affrontare i problemi, piccoli o grandi, che la nostra vita generosamente ci propone. In terzo luogo, sapere e saper risolvere problemi che si rinnovano ogni giorno nel mondo ci rende appetibili alle aziende che hanno problemi ma non hanno “quel” know-how specifico che noi siamo in grado di offrire. In quarto luogo, conoscere il mondo ci aiuta a comprendere noi stessi, non per sostituire il nostro medico di famiglia ma almeno evitarlo il più possibile.

Riguardo al know-how richiesto da una azienda per ricoprire un determinato ruolo è bene specificare che l'azienda non assumerà mai la persona per esigenze di tipo orizzontale, ma generalmente, i requisiti richiesti per un impiego a tempo pieno sono di tipo verticale, ovviamente stiamo parlando di impiegati. Quindi si arriva al paradosso che maggiore è la conoscenza, minori sono le probabilità che una azienda ci assuma, ma sicuramente, saremo appetibili per quell'azienda a livello di consulenza. Ma anche in questo caso il processo di integrazione con una azienda è lungo e laborioso.

Per concludere questo paragrafo oso affermare che maggiore è il livello di conoscenza maggiori sono le probabilità di ricoprire un ruolo autonomo in questa collettività, con tutte le conseguenze, positive e negative,

che ne derivano. E minori saranno le probabilità che una azienda ci assuma alle sue dipendenze.

“Sai, essere liberi, costa soltanto, qualche rimpianto”
Vasco Rossi.

1.3 – Penso dunque sono.

Come facciamo a pensare se non abbiamo gli elementi fondamentali per costruire e valutare i nostri processi mentali? Provate a chiedere ad un muratore di costruirvi una casa però senza mattoni.... E nel costruire la mia casa mi serviranno mattoni piccoli, mattoni medi, mattoni grandi, pezzi di mattoni, è più variato sarà il mio repertorio di materie prime e strumenti più alta sarà la probabilità di riuscire ad esprimere il progetto che avevo in mente. Così è la conoscenza, maggiore è il nostro livello di conoscenza, maggiori saranno le probabilità di riuscire ad esprimere quello che abbiamo dentro, sia questa scienza, tecnica, religione o arte. Creatività non significa creare, solo Dio crea, noi ci limitiamo a stabilire nuove interconnessioni tra elementi conoscitivi sino a quel momento sconosciuti, questo significa creatività. La conoscenza è il cibo della nostra mente, privarsene significa soccombere....

Ma il messaggio di Cartesio andava oltre a questo che ho appena detto, il messaggio di Cartesio significa che l'uomo ha consapevolezza del proprio pensare, a differenza degli animali se si escludono alcuni primati.

“Cosa sarà, che fa crescere gli alberi, la felicità...”

Lucio Dalla, Rosalino Cellamare.

Vorrà dire che se abbiamo “troppa” conoscenza, ci avanzerà! Ma mai, per nessuna ragione, potrà nuocerci!

La conoscenza è come l'olio in un motore, l'amperaggio di una sorgente di corrente elettrica. Quello in più non si usa. Avercene!

Quindi concluderei questo paragrafo con la frase; apprendo quindi sono.

1.4 – I simboli.

A volte noi parliamo di una cosa, e siamo convinti sul significato di ciò di cui stiamo parlando, ma attenzione; più ci spingiamo in alto con l'astrazione e più rischiamo che il contenuto sia camuffato dalla sua definizione. Questo è il potere dei simboli, il polimorfismo. Cioè, una frase, una parola, pronunciati in un contesto assumono valore di verità in un secondo, in un terzo o addirittura in molteplici contesti. Questa caratteristica è nota in informatica, lo stesso numero può essere espresso ad esempio in forma binaria, come 1110, ed allo stesso tempo in forma esadecimale, cioè, E. In questo caso, due simboli diversi 1110 ed E, sono due simboli diversi che rappresentano lo stesso valore; 14.

I simboli hanno un potere enorme, oserei dire magico. Sì, perché se io riuscissi a legare due realtà tramite un simbolo; cosa potrebbe vietare il contrario? E se io manipolando la realtà riesco ad alterare il simbolo che la rappresenta, chi avrebbe l'ardore di affermare che il processo opposto è impossibile?

Nei circuiti elettronici, è noto, la variazione di un simbolo determina la variazione dello stato fisico del circuito. Lo stesso avviene nel nostro cervello; se io penso o compio un atto sessuale i “circuiti elettrici” attivati, e modificati, sono diversi da quelli determinati dall'azione di mangiare un gelato. Addirittura, se io compio una azione nuova, cioè mai effettuata prima, cambia “la struttura” del circuito. Se noi potessimo “creare” nuove situazioni simboliche, ed intendo con questo generare situazioni mai viste, avremmo creato un nuovo circuito, una

appendice, che prima non c'era.

Nella didattica i simboli giocano un ruolo fondamentale, ecco perché la stessa lezione impartita ad alunni diversi lascia impronte diverse in ciascuno di loro. Perché il patrimonio di simboli di ciascuno degli alunni coinvolto ed attivato dalle parole del docente è diverso per ciascuno di loro.

Taluni demonizzano la violenza fisica auspicando in sua vece il colloquio, la negoziazione. In realtà, le parole, possono ferire più della violenza fisica.

Crederne profondamente in una idea ci dà una forza enorme perché ogni successo che avremmo raggiunto lo addebiteremmo a quell'idea rafforzandone l'importanza. Ecco che il simbolo che rappresenta la nostra idea assume un potere di oracolo, indipendentemente dal fatto che la nostra idea sia giusta o sbagliata. Chi può ragionevolmente affermare senza ombra di dubbio che la mucca non sia sacra? Qual è quel uomo che può permettersi di affermare ragionevolmente che un altro uomo sia folle?

1.5 – Patologia dell'insegnamento.

Ciascuno di noi dovrebbe farsi un esame di coscienza almeno una volta al giorno. Toglie il medico di turno. Gli insegnanti dovrebbero farsi un esame di coscienza almeno due volte al giorno. Mi ricordo che anni fa, un mio conoscente, il quale mi aveva fatto ottenere una docenza per un corso di formazione professionale, al mio interrogarlo sul come fossero andate le mie lezioni mi rispose : “Eh... stai un po' troppo in cattedra...”. Il che stava a significare che il mio atteggiamento nei confronti degli alunni era un po' troppo dall' alto verso il basso. Quanto son grato a questa persona!! Docenti! Non sentitevi in cattedra come Dio nel tempio! Lo dico per il vostro bene. Noi siamo un insieme sinergico con i nostri alunni, siamo docenti perché qualcuno ha la voglia di ascoltarci. Non dimentichiamocelo. È un gioco di ruolo, in quel momento il flusso della conoscenza scorre dal docente verso i suoi alunni ma gli alunni insegnano nel contempo al docente il ruolo di insegnante. Io, perfetto asociale, ma grande filantropo, penso che la più grave patologia di cui un docente, un genitore, un manager possano essere affetti è lo stare in cattedra.

Facile, direte voi, ma l'insegnante, il genitore, il manager sono io ! Ho delle responsabilità!! Per questo mi pagano! Appunto; per questo vi pagano.

1.6 – Cosa farai da grande?

“Mio figlio ha cinque anni e cinque convinzioni, facendo bene i conti, ne ha cinque più di me”

Gino Paoli.

Cosa farai da grande? È la domanda che ognuno di noi si è sentito rivolgere più di una volta quando era piccolo, di solito alle scuole medie, ma ricorrono episodi in intervalli temporali anche più ampi. Alla domanda rivolta ad Edoardo de Crescenzo su quale fosse il suo mestiere in una età già sopra la maturità egli ha risposto; “I’m an unemployed engineer”, sono un ingegnere disoccupato. Da adolescente un prete mi chiese cosa mi sarebbe piaciuto svolgere come mestiere una volta che fossi divenuto grande. Io risposi; il consulente. Il prete sorrise e replicò; la più alta forma espressiva e libera del lavoro; la libera professione. Ma consulente di cosa? Se ancora ero all’ABC dell’istruzione e di certo non avevo ancora un mestiere in mano ne in incubatrice? Devo dire che la mia risposta risentiva in qualche misura del lieve plagio indotto in me dal mio professore di italiano e storia delle scuole medie Umberto Spadoni, ma non aveva sbagliato tanto nel vedere in me uno spirito libero.

Come può un bambino capire cosa gli piacerà divenire quando sarà grande? Questa è l’immensa responsabilità di un educatore, un insegnante; orientare l’alunno verso le proprie inclinazioni, attitudini, aspirazioni. E questo è un lavoro che dura tutta la vita. Grazie a Dio, avremo

sempre dei maestri durante l'arco di tutta la nostra esistenza. Ed i nostri maestri saranno giusti maestri, falsi maestri, allievi maestri, maestri allievi. Nei corsi per il "successo nella vita" vi insegneranno a sforzarvi di ricordare quali fossero i vostri sogni e aspirazioni quando eravate bambini, ancora incontaminati dai pareri dei Maestri. Se qualcuno di noi ha avuto la fortuna di anelare ad una aspirazione incontaminato dai pareri degli altri quando era bambino ha buone probabilità di trovare lì la realizzazione del proprio essere. Ma nella vita si cambia anche, veramente, radicalmente. Siamo liberi alla sorgente o alla foce del nostro percorso, se mai saremo liberi?

Sarebbe bello svegliarci una mattina da cinquantenni che eravamo a bambini di dieci anni, quante gliene diremmo a quel signore di mezza età che ha frantumato i nostri sogni, venduto i propri ideali per qualche banconota, rinnegato le nostre puerili sì ma aspirazioni, messo al mondo dei figli come noi? Che delusione....

Ci piaceva tanto il nostro papà e adesso lo abbiamo rimpiazzato divenendo papà noi. Ma noi eravamo piccoli! Docenti ricordate sempre ai vostri alunni i bambini che erano, fate uscire da loro lo spirito che gli anni, le delusioni, i compromessi della vita, hanno costretto a rinnegare.

1.7 – Psicologia della didattica.

Il termine psicologia deriva dal greco e significa studio dell' anima, dello spirito. Cosa è che ci conduce dopo un lungo percorso di studi a diventare noi stessi insegnanti? Io penso che tra i sensi di gratitudine più grandi che io abbia mai provato nella mia vita ci sia la mia gratitudine nei confronti dei miei insegnanti che hanno saputo in primo luogo trasmettere in me degli insegnamenti che difficilmente avrei altrimenti potuto acquisire ed in secondo luogo quegli stessi insegnanti mi hanno trasmesso l'amore per il sapere ed il tramandare il sapere in un atto che andava al di là delle cognizioni trasmesse e che era un mezzo per amare, prendersi cura di me, dimostrando quanto essi tenessero a me. Questo sia in ambito scientifico che nello sport. È un po' come l'istinto genitoriale che ci viene trasmesso dai nostri genitori, ma non tutti diventiamo a nostra volta genitori. Amiamo perché siamo stati amati, insegniamo perché ci è stato insegnato. Altre volte abbiamo talmente tante nozioni ed idee per la testa che abbiamo voglia di comunicarlo al mondo intero per metterci in discussione e confrontarci. Non dimentichiamo che la critica dei nostri alunni è implacabile. Un'altra componente purtroppo presente in misura diversa in ciascun docente è lo stare in cattedra; “Finalmente ora dovranno ascoltare ciò che ho da dire; nolenti o dolenti!”, capita anche questo. Nei college americani i docenti vengono sottoposti a test di gradimento da parte degli allievi, e la direzione tiene conto seriamente di tali valutazioni. In Italia è molto frequente il connubio insegnamento-libera professione

che consente, soprattutto all' inizio della carriera di disporre di un reddito per consentirci di avviare la nostra attività libero professionale ma nello stesso tempo arricchisce il nostro portafoglio di esperienze professionali aumentando il nostro valore di insegnanti. Capita anche che si finisca per insegnare perché è l'unico lavoro che ci consentono di effettuare, speriamo per merito e non demerito.

“Chi sa fa, chi non sa insegna”, l'avete mai sentita pronunciare questa frase? A me, onestamente, fa un po' ridere... ma purtroppo in certi casi assume valore di validità.

1.8 – Mi faccia un esempio.

L'esempio è un esercizio in cui noi applichiamo le conoscenze acquisite per risolvere un problema o applicare una metodologia. Nelle materie cosiddette scientifiche l'esempio, l'esercizio sono procedure fondamentali per stabilire la riproducibilità della metodologia, del paradigma.

Nell'esempio, nell'esercizio noi dimostriamo di aver compreso la teoria. A parole siamo bravi tutti ma quando si tratta di dare una dimostrazione alle nostre asserzioni molto di noi cadono. E chi sa fare ma non conosce la teoria? In questo caso l'interessato svolge una attività manualistica, cioè da manuale, esegue delle istruzioni che qualcuno gli ha tramandato. Ma un conto è eseguire istruzioni e un conto è idearle; questa è la differenza tra la teoria e la pratica. Il perito applica le istruzioni e le formule scritte sul manuale, l'ingegnere scrive il manuale. Quindi un po' di rispetto per chi lavora di teoria per favore. Il rischio di chi pratica teoria è di effettuare voli pindarici, uscire dal seminato, ma non è forse soggetto ad errori umani l'operato di un operaio?

Il caro babbo di un mio amico, docente universitario, quando ero ragazzo mi fece un esempio; la società è come un alveare, ci sono le api operaie, le api soldato, le api impollinatrici, l'ape regina, l'alveare per sopravvivere necessita di ciascuna di queste api; tu che ape vuoi essere? Una singola idea può dare lavoro a migliaia di individui, non dimentichiamocelo. Il fatto è che le idee, quelle buone, non vengono a comando, ma sicuramente lo studio aiuta. La ricerca è l'unica strada che conduce a

nuove idee. È chiaro, il più semplice degli operai può partorire una idea rivoluzionaria, ma anche qui bisogna distinguere tra idee che rivoluzionano le tasche di pochi dalle idee che rivoluzionano positivamente l'intera collettività.

È noto che quando presentiamo una idea all'ufficio brevetti per prima cosa una commissione stabilisce se l'idea abbia un interesse nazionale o militare, poi eventualmente viene concesso il brevetto.

“Guarda che ti faccio l'esempio!.... Sua moglie è rimasta in cinta! “

Cochi e Renato

1.9 – Ambasciator non porta pene.

Chi è l'ambasciatore in tutto questo contesto ?
L'ambasciatore è la Didattica: l'insegnamento ed i metodi relativi.

“La **didattica** (dal [greco](#) *διδάσκω*, cioè "insegnare"), indica la [teoria](#) e la pratica dell'insegnamento. Si può dividere in didattica generale, che riguarda i criteri e le caratteristiche generali della pratica educativa, e in didattica speciale che riguarda invece i singoli insegnamenti o le diverse caratteristiche (età, capacità specifiche, ambiente) dei soggetti dell'apprendimento.” (Wikipedia)

In quanto tentativo di trasmettere conoscenza da una sorgente, l'insegnante, ad uno o più riceventi, gli alunni; la Didattica è per definizione esente da vizi. Un difetto di comunicazione si trasforma automaticamente in una opportunità di miglioramento del metodo di insegnamento ed è esso stesso oggetto della Didattica, materiale che non può che alimentare questo ramo della scienza. La Didattica nell'insegnare corregge se stessa, svolgendo la sua funzione. Il materiale della Didattica è l'insegnamento ed ogni evento si trasforma in opportunità di miglioramento del metodo.

Capitolo 2 - Esempi

2.1 – Proviamo ad insegnare.

In questo capitolo proverò a dare un esempio di insegnamento. Visto che la mia specializzazione è la informatica, tenterò di fornire un esempio di docenza con commenti e critiche per la materia di informatica. Ma mi limiterò ai concetti base, affinché la lettura possa essere alla portata di tutti. Proverò ad esporre i concetti basilari dell'informatica che seguono:

i-Informazione.

ii-Informatica.

iii-Computer.

2.2 – Informazione.

Il docente potrebbe ad esempio introdurre il concetto di informazione con la seguente definizione:

“L’informazione è una relazione tra due dati che diminuisce il grado di incertezza della nostra conoscenza“

Nel dare una definizione noi usiamo dei termini che potrebbero essere noti, sconosciuti o dubbi alle orecchie dei nostri interlocutori. Si innesca così una ramificazione ad albero che ha come radice, cioè punto di partenza, la nostra definizione. La radice, cioè la definizione, ha la responsabilità di dirigere l’evoluzione di tutta la discussione dal suo inizio. L’evoluzione della discussione dipenderà da quali siano i connettori semantici in possesso di ciascun ascoltatore e dai loro feedback. Ad esempio un primo alunno potrebbe chiederci; che cosa è una relazione? Un secondo alunno potrebbe chiederci; perché? Ve lo ricordate il gioco amato dai bambini quando raggiungono una certa età? Il gioco dei perché. Qualunque cosa voi risponderete vi viene replicato con un “perché?”. È un gioco che potenzialmente non ha fine, si potrebbe arrivare all’atomo o addirittura a Dio. Sta al docente “placare” con informazioni opportune quegli stati di perplessità indotti dalle proprie parole nella psiche di ciascun alunno. Non si può saper tutto, nemmeno il docente ha questo dono. Ma si può fornire una spiegazione “esaudiente”, cioè, sono in grado di capire, ripetere ed applicare quello che il docente ha spiegato. Capire al cento per cento quello che un nostro interlocutore sta cercando di illustrarci è impossibile,

dovremmo avere il suo stesso cervello ed il suo stesso vissuto. Esistono invece dei “livelli” di comprensione che sono adeguati quando io sono in grado di applicare quelle nozioni ricevute nella pratica, l’esercizio.

Per non parlare delle evocazioni emotive. Ad esempio una persona che non sappia cosa sia una relazione in senso informatico la prima cosa a cui potrebbe essere indotto a pensare nell’udire questo termine potrebbe essere la sua amante. Ma l’apprendimento è fatto anche di queste associazioni emotive.

2.3 – Informatica.

Il docente potrebbe ad esempio introdurre il concetto di informatica con la seguente definizione:

“L’informatica è la scienza che si occupa dell’ordinamento, trattamento e trasmissione dell’informazione tramite elaboratore elettronico.”

Qui vediamo un esempio di definizione che si basa su di un'altra definizione fornita in precedenza; l’informazione. Quando il docente fornisce una definizione basata su un'altra definizione fornita in precedenza deve aver dato il tempo all' alunno di aver “digerito” la definizione precedente. Un altro aspetto da evidenziare delle definizioni è quello che per alcune materie queste subiscono delle evoluzioni nel tempo. Ad esempio il termine informatica è partito dal significato esposto dalla sua etimologia; informazione automatica, sino ad arrivare a parlare di “trattamento”, termine che assume connotati ben più ampi ed include significati in ambito sociale con tutte le conseguenze che ne derivano. D’altra parte come non avrebbe potuto una materia che si occupa di informazione non includere argomenti di comunicazione tecnicamente parlando ma anche da un punto di vista sociale? Quindi in sintesi possiamo dire che l’informatica si occupa della gestione della informazione mediante ausilio di sistemi per l’elaborazione dell’informazione; gli elaboratori elettronici. Vediamo allora anche come l’aspetto storico di una materia di insegnamento assuma un ruolo che oserei definire importantissimo. Perché la storia ci spiega le motivazioni delle ricerche delle scoperte fungendo da collante per il

nostro processo di apprendimento delle informazioni.

2.4 – Computer.

Il docente potrebbe ad esempio introdurre il concetto di computer con la seguente definizione:

“Apparecchio elettronico in grado di effettuare calcoli, eseguire operazioni sui dati e gestire grandi quantità di memoria ad una velocità estremamente più rapida di quella accessibile ad un essere umano....”

In questo caso la definizione fornita dal docente coinvolge contenuti tecnologici. La tecnologia non è un'opinione, questo è apparso abbastanza chiaro all'uomo sin dalla sua prima scoperta e relativa capacità di controllarla: il fuoco. La scienza ci aiuta a sviluppare la tecnologia e la tecnologia ci fornisce nuovi strumenti, nuovi approcci per sviluppare nuova conoscenza. È un sistema a retroazione che prende il nome di progresso. In generale il progresso serve a migliorare la condizione dell'uomo, ma non tutti siamo d'accordo con questa ultima affermazione. L'uomo ha messo piede sulla luna, ha inviato robot su Marte, ha oltrepassato i confini del sistema solare con le sue sonde, ha oltrepassato i confini della galassia con i suoi telescopi, ha infine oltrepassato i confini dell'universo con la sua immaginazione; il suo intelletto.

Capitolo 3 – La programmazione

3.1 – Un po di storia.

Nel 1936 un matematico britannico di nome Alan Turing ideò un modello per eseguire calcoli ed elaborazioni in modo meccanico noto come Macchina di Turing. Tale modello diede vita ad una tecnologia tutt' oggi in via di sviluppo ed alla ricerca di una teoria unificante; la programmazione.

Negli anni quaranta i primi programmi per computer venivano scritti in linguaggio macchina su elaboratori che occupavano stanze intere. Il linguaggio macchina non è altro che il codice binario, per intenderci, il numero, che rappresenta all'interno dell' elaboratore, una istruzione.

La macchina di Turing non è altro che un nastro di lunghezza teoricamente illimitata con una finestra attraverso la quale noi possiamo leggere e scrivere sul nastro; questo è un programma.

Dagli anni quaranta ad oggi si è via via passati dal linguaggio macchina ai linguaggi di programmazione, via via, si può dire ?, più evoluti. Il parametro che ha contraddistinto questa evoluzione è stato via via un aumento del grado di astrazione. Se nel linguaggio macchina ogni istruzione rappresentava una singola operazione dell'elaboratore, con l'astrazione e l'associato aumento della potenza di calcolo e della integrazione dei circuiti elettronici, una singola istruzione di un moderno linguaggio evoluto determina l'esecuzione di molteplici istruzioni macchina.

È avvenuto, per i linguaggi di programmazione, quello

che un po' avviene nel nostro cervello nell' arco del periodo evolutivo dell' apprendimento; si parte dai mattoni, dalle basi, per arrivare, con una parola, a racchiudere una storia intera.

Non starò ad elencare i vari linguaggi che si sono susseguiti sequenzialmente e spesso parallelamente nel corso degli anni, sarebbe un tecnicismo. Ma la maggior astrazione via via raggiunta dai vari linguaggi di programmazione ha consentito di risolvere problemi di ingegnerizzazione, produttività, modellazione, sicurezza; ma allo stato dell'arte il percorso dei linguaggi di programmazione si trova di fronte lo stesso ostacolo di cui è vittima la burocrazia dei sistemi moderni; l'eccessiva astrazione non riesce ad avere controllo sul singolo dettaglio, e così, un semplice dettaglio, può mandare in crisi un intero sistema.

3.2 – La mia storia di programmazione.

Ho iniziato ad interessarmi di programmazione non appena entrato in possesso del mio primo computer, un Olivetti M20. Il personal computer Olivetti M20 è stato uno dei primissimi computer disponibili al grande pubblico nei primi anni ottanta, insieme all'Amiga, all'Apple ed al Lemmon, per intenderci. Mi ricordo, allora avevo sedici anni, che il suo costo era l'equivalente di un'automobile; infatti, mio padre, fece un leasing per acquistarlo. Subito imparai a scrivere programmi e a svolgere qualche lavoretto commissionatomi dal titolare del concessionario Olivetti di zona Silvano Bellazzecca. Nel frattempo stavo frequentando l'ITIS "E.Mattei" di Urbino indirizzo elettronica industriale ed ebbi modo di approfondire alcuni aspetti della programmazione grazie ad alcuni miei docenti, tutti ingegneri elettronici, tra cui cito in particolare il caro Ing. Giuseppe della Chiara che ho ancora saltuariamente modo di incontrare per motivi musicali. Peppe mi chiamava "little white"..... Grazie Peppe.

Una volta diplomato come perito elettronico mi sono iscritto alla facoltà di ingegneria elettronica all'università degli studi di Bologna e qui ho avuto modo di affinare la mia teoria nei confronti dell'informatica. Mi ricordo che già studente di ingegneria ero titolare di partita Iva e svolgevo qualche lavoretto di programmazione. Addirittura mi proposero di abbandonare gli studi di ingegneria per lavorare come programmatore dipendente presso una azienda a cui avevo effettuato un lavoro. Non so se ho fatto bene a rifiutare.... Ancora oggi me lo

chiedo... Infine, una volta laureato in ingegneria elettronica ho iniziato quasi subito a lavorare, il mio primo lavoro, a meno di un mese di distanza dalla laurea è stato quello di insegnante di elettronica presso l'IPSIA "G.Benelli" di Pesaro e, contemporaneamente ho iniziato, anzi proseguito, il mio lavoro di programmatore che mi ha accompagnato per gran parte della mia carriera lavorativa. Attualmente continuo a programmare ma non a tempo pieno, on demand e contemporaneamente insegno, scrivo e faccio video su Youtube.

3.3 – Che cosa è un programma?

Si. Va bene, la macchina di Turing, Ma che cosa è un programma? La domanda potrebbe sembrare banale, ma non lo è affatto, almeno per la maggioranza delle persone che stanno leggendo questo libro. Avete presente un documento? Sì, un documento, tipo una lettera di quelle che il postino vi imbuca nella cassetta delle lettere di tanto in tanto. Una lettera riporta un testo, ciò che chi vi scrive ha intenzione di comunicarvi nell'inviarvela. E voi cosa fate? Aprite la busta, spiegate la lettera e cominciate a leggere. Successivamente, dopo aver letto la lettera, in genere eseguirete dei compiti in funzione di ciò che avete letto nella lettera, che saranno; pagare una bolletta, pagare una multa, o semplicemente rispondere con un'altra lettera a chi vi ha inviato la sua. Ecco, un programma è un documento, una lettera che noi inviamo al computer, più precisamente al microprocessore del nostro computer, affinché questi esegua delle operazioni. E più precisamente il programma è un insieme sequenziale di istruzioni che noi inviamo al nostro microprocessore affinché questi svolga delle operazioni; cioè, scriviamo sul nastro della nostra macchina di Turing. E per scrivere questo elenco di istruzioni da inviare al nostro microprocessore avremo bisogno di un linguaggio che sia comprensibile al nostro computer, che lo tradurrà in linguaggio macchina e a noi che lo dobbiamo utilizzare per scrivere il nostro programma. Esistono diversi tipi di linguaggio di programmazione in funzione del compito che il programma dovrà assolvere. Non solo, col trascorrere del tempo vengono ideati nuovi

linguaggi, alcuni vengono rimpiazzati, altri, per le loro peculiarità, resistono ai nuovi arrivati mantenendo la loro nicchia di campo di applicazione. Non per niente il termine linguaggio deriva da lingua e cioè insieme di termini e regole atte a comunicare, con la lingua comunichiamo tra esseri umani, con un linguaggio di programmazione comunichiamo da essere umano a macchina, il computer, indicandogli quali sono le operazioni che deve eseguire.

3.4 – I linguaggi di programmazione.

Credo che l'argomento linguaggi di programmazione meriti un paragrafo tutto suo. Abbiamo già avuto modo di appurare come il primo linguaggio di programmazione per elaboratori sia stato il linguaggio macchina. Ma vediamo un esempio di cosa significhi macchina è cioè nel nostro caso microprocessore, la nostra macchina di Turing. Immaginiamo di avere una scatola magica che effettua operazioni matematiche; ora immaginiamo che questa scatola magica, al pari di una cassetta delle lettere, abbia una feritoia dalla quale noi possiamo inserire dei fogli con sopra scritto cosa deve fare la scatola magica. Ad esempio se volessimo effettuare la somma di due numeri, come $2+3$, potremmo inserire nella feritoia della nostra scatola magica un primo foglio con su scritto 2. Successivamente inseriremmo nella feritoia un secondo foglio con su scritto 3. Infine dovremmo dire alla scatola magica; esegui la somma dei due numeri che ho inserito con i fogli; quindi, a questo scopo potremmo inserire il terzo ed ultimo foglio con su scritto 'più'. A questo punto la scatola magica ci potrebbe rispondere facendo uscire un foglio dalla feritoia con su scritto 5.

Un microprocessore questo fa, esegue delle somme, delle sottrazioni, moltiplicazioni e divisioni. 'più' è l'istruzione macchina che noi, grazie al nostro linguaggio macchina, impartiamo al nostro processore. Ed il nostro linguaggio macchina comprenderà anche altre istruzioni, come 'meno', 'per', 'diviso'. E se noi inventassimo un ufficio che raccoglie lettere di chi non conosce il linguaggio macchina ma vuole scrivere al processore, questo ufficio

si occuperebbe di tradurre dal linguaggio naturale dei suoi interlocutori in linguaggio macchina da imbucare nella feritoia della nostra scatola magica. In questo caso il linguaggio naturale dei clienti dell' ufficio andrebbe a costituire il linguaggio di programmazione del nostro processore.

Da quest' ultima rappresentazione possiamo dedurre che un linguaggio di programmazione è uno strumento che ha come mattoni costituenti le istruzioni macchina, è una piramide di astrazioni che si ergono sul linguaggio macchina. Una istruzione di un linguaggio di programmazione evoluto determina l'esecuzione di innumerevoli istruzioni in linguaggio macchina. A partire dagli anni quaranta sino ad oggi si sono susseguiti molteplici linguaggi di programmazione, ne cito alcuni; Fortran, Pascal, Ada, Smalltalk, Basic, Cobol, C, C++, C#, Java, Lisp, Prolog, per passare ai linguaggi di programmazione in ambiente Web come HTML, Asp, PHP, JSP, Python, e via dicendo...

Interessante la classificazione che gli studiosi, ma che è anche una conseguenza naturale, hanno fatto di tutti i linguaggi di programmazione sino ad oggi ideati, e cioè; imperativi e dichiarativi. I primi, quelli imperativi, sono quei linguaggi in cui le istruzioni altro non sono che ordini al processore, direttive; fai questo, prendi quell' altro, esegui. Mentre nei linguaggi dichiarativi non vengono impartiti dei comandi ma vengono dichiarate delle definizioni che l'interprete del linguaggio elaborerà determinando delle reazioni. La maggior parte dei linguaggi sino ad oggi ideati sono di tipo imperativo, pochi altri come Lisp e Prolog sono di tipo dichiarativo.

Ora qui esporro' una mia opinione, quindi prendetela come tale; i linguaggi dichiarativi sono una mera illusione, infatti sotto l'interprete del linguaggio dichiarativo i fatti si risolvono in comandi imperativi, perché la macchina di Touring questo è. Quindi secondo me il linguaggio dichiarativo è una chimera. A questo proposito è bene citare che qualche anno fa vennero annunciati microprocessori di quinta generazione basati su una programmazione dichiarativa come Prolog e Lisp, in particolare furono le nazioni orientali a preannunciare tale generazione di processori, ma purtroppo non se ne è saputo più niente, come la fusione fredda.... Che questa tecnologia sia finita nelle mani di qualche organizzazione segreta o dei militari o dell'intelligence o della Korea del Nord?

La sfida più grande dell'informatica è la programmazione attraverso il linguaggio naturale, con le sue contraddizioni, paradossi, poemi e sentimenti, di questo se ne occupa una branca dell'informatica che si chiama intelligenza artificiale. Infine per terminare questo paragrafo, parlando di macchine di Touring, linguaggio macchina e relativi codici binari, è il caso di citare il relativamente recente arrivo in casa; il qbit. Se i processori ed i relativi elaboratori degli anni quaranta erano basati sul bit, unità di informazione che può assumere due valori; zero o uno, il qbit è una unità di informazione quantistica rappresentata da un vettore nello spazio di Hilbert ed è una combinazione tra i due stati zero e uno. In soldoni, un singolo qbit, a differenza di un bit che può assumere due valori, può rappresentare tutto lo scibile umano.

3.5 – Dati e funzioni.

Per entrare più nel dettaglio, diciamo nell' architettura di un programma, possiamo dire che molto genericamente un programma è costituito da un insieme di dati ed un insieme di funzioni che operano su di essi. Per esempio potremmo avere un programma che chiameremo `BilancioMensile`, che avrà come dati l'elenco degli importi dei prodotti che abbiamo acquistato in un mese; e come funzioni tutte le operazioni di calcolo per determinare le varie voci di bilancio della nostra spesa. Così ad esempio potremmo avere la funzione `SpesaMediaGiornaliera`, la funzione `SpesaTotaleMensile` e via dicendo. Le funzioni anche conosciute come procedure o subroutines altro non sono che un insieme di istruzioni, racchiuse in una porzione di programma, che assolvono ad un determinato compito. Ogni volta che avremo bisogno di assolvere a quel determinato compito non faremo altro che evocare quella determinata funzione. È chiara l'utilità di una funzione, infatti questa potrà essere evocata in più punti diversi del programma senza avere la necessità ogni volta di riscrivere tutto il codice, riducendo la probabilità di errori e aumentando la manutenibilità del codice.

In un programma i dati possono essere di tipo diverso, così avremo ad esempio, per citarne alcuni; i numeri interi, i numeri reali, le stringhe o testi, le date, i numeri logici o booleani, e via dicendo. Insieme ai dati troveremo le strutture di dati che sono insiemi che contengono diversi dati di tipo diverso tra loro. Il tipo di dato identifica univocamente i valori che esso può assumere e

le operazioni che possono essere effettuate su di esso.
Vediamo un piccolo esempio:

```
programma esempio ()
```

```
{
```

```
    Intero i;
```

```
    i=10;
```

```
    incrementa_i ();
```

```
// commento : a questo punto del programma i vale 11
```

```
    fine();
```

```
funzione incrementa_i(
```

```
{
```

```
    i=i+1;
```

```
}
```

```
}
```

3.6 – La programmazione ad oggetti.

Sicuramente la più interessante ed utile innovazione introdotta nel corso degli anni di evoluzione dei linguaggi di programmazione è stata il paradigma della programmazione ad oggetti (OOP - Object Oriented Programming)

Un programma OOP è organizzato utilizzando una entità che prende il nome di classe. La classe altro non è che una struttura contenente dati e funzioni. Non entrerà nel dettaglio ma vediamo quali sono le tre caratteristiche fondamentali che caratterizzano l'uso delle classi:

- i. Incapsulamento.
- ii. Ereditarietà.
- iii. Polimorfismo.

Per incapsulamento si intende la proprietà per cui una classe ha la capacità di racchiudere e nascondere al resto del mondo i suoi dati e le sue procedure interni. Per comunicare ed asservire ai suoi compiti una classe esporrà solo i metodi e le proprietà strettamente necessari. La classe opera un po' come una scatola nera; l'utilizzatore sa cosa fa ma non sa come lo fa.

Attraverso l'ereditarietà noi possiamo progettare una nuova classe ereditando tutte le proprietà definite in una seconda o molteplici classi. La nuova classe avrà tutte le capacità della o delle classi ereditate ed in più potrà personalizzarne metodi e proprietà o addirittura ovviamente crearne di nuovi. È chiaro che una tale

caratteristica produce una nuova abilità a processi di ingegnerizzazione, manutenibilità, e produttività.

Il polimorfismo è una proprietà di non immediata comprensione, molti programmatori lo utilizzano ma non saprebbero spiegarlo. A questo scopo immaginiamo una chiave inglese che riesca ad adattarsi alla dimensione dei bulloni; non avremmo bisogno di cambiare chiave al variare della dimensione dei bulloni; cioè potremmo usare la stessa chiave per ogni dimensione dei bulloni. Questo è il polimorfismo; in termini informatici è la proprietà secondo cui con la stessa classe o metodo, possiamo operare su tipi di dati diversi.

Sinceramente il paradigma OOP è stata un po' una certificazione, formalizzazione, di caratteristiche che i programmatori più avveduti adottavano naturalmente. La scienza a volte ha questo gusto di scoprire l'acqua calda. Ma ovviamente scherzo... In una ottica di ingegneria del software, questo paradigma ha reso possibili progressi in ambito informatico altrimenti difficilmente raggiungibili.

3.7 – Gli algoritmi.

Parallelamente alla ingegneria del software, grazie all'avvento degli elaboratori e dei linguaggi di programmazione sempre più evoluti, si è sviluppata una nuova branca della matematica, la matematica applicata, che trova uno dei suoi campi di applicazione nella ricerca operativa e che consiste nella risoluzione di problemi matematici attraverso l'ausilio degli elaboratori elettronici. La matematica applicata, la ricerca operativa, non richiedono di essere degli esperti in tecnologie informatiche, se non come semplici utilizzatori. L'obiettivo è il problema matematico che viene risolto attraverso un programma che mette in opera un algoritmo.

La definizione di algoritmo è la seguente: “Numero finito di passi da intraprendere per risolvere un problema “.

Ma vediamo subito un esempio di un semplice algoritmo; il nostro algoritmo sarà così definito dai seguenti passi:

Algoritmo: “Portare a spasso il cane”

1. Prendere due sacchetti per le feci.
2. Prendere la paletta per raccogliere le feci.
3. Prendere il guinzaglio.
4. Applicare il guinzaglio al cane.
5. Uscire di casa con il cane e chiudere la porta.
6. Iniziare a passeggiare.
7. Il cane è andato di corpo? Se sì vai al punto 8. Altrimenti vai al punto 7.
8. Raccogli le feci con la paletta ed introducile in un

sacchetto.

9. Al primo bidone deposita il sacchetto con le feci.
10. Vuoi tornare a casa? Se si vai al punto 11 altrimenti continua la passeggiata e vai al punto 10.
11. Incamminati verso casa.
12. Quando arrivi a casa apri la porta, toglì il guinzaglio al cane e riponilo nel suo alloggio.
13. Fine algoritmo.

Ovviamente questo è un banale esempio e può sicuramente essere migliorato ma rende l'idea di cosa sia un algoritmo. L'uomo ha sempre adottato algoritmi da quando è apparso sulla terra, solo che non li chiamava così.

Per concludere vediamo un algoritmo serio atto alla risoluzione di un semplice problema : determinare il più grande tra due numeri.

Algoritmo : determinare il maggiore tra i numeri A e B

1. A è maggiore di B ? se si vai al punto 3
2. B è maggiore di A ? se si vai al punto 4
3. Stampa : "A è maggiore di B". Vai al punto 5
4. Stampa : "B è maggiore di A".
5. Fine algoritmo.

3.8 – Differenza tra una App e una WebApp.

Che differenza c'è tra una App che viene eseguita sul nostro computer e una WebApp cioè quelle App che eseguiamo per mezzo di un browser Internet? In tutti due i casi è sempre un processore, eventualmente più di uno, che esegue l'App o la WebApp, solo che nel caso della App il processore e quindi il computer che esegue l'App risiedono nello stesso computer, mentre quando noi eseguiamo una WebApp questa viene eseguita in computer diverso da quello dove la stiamo visualizzando attraverso il nostro browser. Tutto ciò comporta che una WebApp richieda un linguaggio di programmazione idoneo a tale scopo che sarà diverso dal linguaggio di programmazione che utilizzeremo per sviluppare una normale App.

Uno dei primi, più noti e semplici linguaggi di programmazione per contenuti web è l'HTML. HTML sta per Hyper Text Markup Language, e cioè linguaggio a marcatori per gli ipertesti. HTML è molto semplice, consente di collegare pagine web tramite links, collegamenti, semplicemente cliccando su una parola come potrebbe essere 'informatica', viene indirizzata la visualizzazione su di un'altra pagina che, sensatamente, fornirà dei dettagli sul concetto di informatica. Altri linguaggi di programmazione Web, come asp, forniscono potenzialità più avanzate, hanno come cornice sempre HTML, ma riescono a svolgere operazioni 'lato server' ben più complesse ed avanzate di quanto non possa fare un semplice programma HTML. Infatti con asp noi possiamo collegarci con un database, eseguire operazioni

locali al server, anche sul sistema e tutto ciò che ne consegue. Ovviamente asp è stato uno dei primi linguaggi web lato server, ne sono seguiti molti altri, Asp.net, PHP, JSP e via dicendo.

3.9 Big data.

Dall'avvento dei computer una delle problematiche che ha richiesto parecchie risorse per la sua gestione è stata la gestione dei dati. Sino all'avvento di internet ma anche in anni successivi le grandi quantità di dati venivano gestite da applicazioni software specializzate in questo; i database. Per citare i più famosi, Sql Server, MySql, Oracle. All'inizio i database non esistevano e bisognava gestire tutto in file separati o alla meglio in un unico file indicizzato. Ma l'evolversi di internet ci ha messo via via di fronte alla problematica di gestire moli di dati enormi, si parla dell'ordine dello Zettabyte, ovvero, miliardi di terabyte. E' noto che un normale sistema operativo e database annesso per computer sia in grado di gestire moli di dati dell'ordine del terabyte. Non solo, si parla di una eterogeneità dei dati veramente vasta, si tratta di collegare immagini, testi, informazioni, musica, presenti nella rete in misura enorme. Questo richiede nuove tecnologie, il classico database non è più sufficiente, occorrono nuovi strumenti, questa è la sfida dei Big data. Solo una elaborazione distribuita su più server a loro volta multiprocessore potrà far fronte alla complessità ed enormità del problema.

3.10 – La teoria della complessità.

La teoria della complessità è una branca della ricerca operativa che si occupa di quantificare la complessità di un algoritmo. La complessità di un algoritmo viene misurata in funzione del numero di passi necessari a risolvere un determinato problema da parte di un algoritmo. In genere la complessità di un algoritmo viene espressa in funzione della dimensione dei suoi dati di ingresso. Così si dirà che un algoritmo ha complessità $O(n)$, cioè proporzionale ad n , se il tempo necessario per la sua esecuzione è proporzionale a n , dove n è il numero di dati del problema. Si è dimostrato che alcuni problemi sono intrinsecamente più complessi di altri, alcuni addirittura che richiedano un tempo di elaborazione infinito per la loro soluzione. Così sono state individuate due classi di problemi, quelli polinomiali, e quelli non polinomiali, cioè quelli che richiedono un tempo polinomiale per essere risolti e quelli che invece richiedono un tempo esponenziale. Un problema tutt'ora aperto è dimostrare se esista un modo per concepire un algoritmo che sia in grado di risolvere in un tempo polinomiale un problema di complessità esponenziale. Cioè, è possibile che i problemi siano tutti facili da risolvere o esistono problemi più difficili che con l'attuale matematica non riusciremo mai a risolvere in un tempo polinomiale ?

3.11 – Colleghiamo un po le cose.

Devo l'esistenza di questo paragrafo al mio amico Franco Marino, esercente di una attività di ristoro, (Franco è un eccellente pizzaiolo), in Pesaro e mio amico da tantissimi anni, ho voluto che lui avesse una copia del mio libro 'la programmazione', riportato in questo libro, in quanto non addetto alla disciplina mi fornisse delle indicazioni sull'efficacia del mio intento divulgativo nello scrivere il libro. Franco mi ha fatto capire che in teoria lui poteva anche seguire la spiegazione di cosa sia un programma, ma non capiva come fosse possibile 'la comunicazione', lui ha detto, tra il documento, cioè il programma e l'interno del computer cioè il cervello, l'hardware. Nel ringraziarlo per questa sua osservazione cercherò di fornire spiegazioni al riguardo nel presente paragrafo.

Ogni computer ha un monitor, uno schermo per intenderci, da lì noi possiamo visualizzare l'anima e la maschera della nostra macchina. La catena trasmissiva dal microprocessore al video e viceversa è, almeno nei computers tradizionali, completamente elettrica. Il microprocessore è un circuito integrato, un chip, dicono in America, quelli dalla forma rettangolare con tanti piedini che vengono saldati sul circuito stampato che costituisce la scheda madre del nostro computer. Attraverso la scheda madre, tramite le piste del circuito stampato, sono collegate al microprocessore, memoria e periferiche come il video, o monitor, la tastiera, e la stampante, nonché l'hard disk, cioè il disco rigido che è una memoria detta di massa con elevate capacità di

memoria persistenti anche in assenza di alimentazione elettrica del circuito.

Attraverso la tastiera possiamo digitare parole o frasi che comportano delle reazioni del computer, come ad esempio far partire l'esecuzione di un programma, questo lo si può ottenere anche cliccando con il mouse su di una icona associata al programma che vogliamo eseguire. Così ad esempio noi possiamo lanciare l'esecuzione di un programma che serve per scrivere programmi, analogamente ad un word processor come word della Microsoft, noi possiamo utilizzare un programma che ci consente di scrivere programmi, non solo, una volta scritto il programma è possibile memorizzarlo sul disco, oppure convertirlo a sua volta in un programma mediante un processo detto di compilazione e successivamente memorizzarlo nel disco. Il sistema operativo del computer, ad esempio Windows, è quello che si occupa di trasferire, quando noi clicchiamo sull'icona del nostro programma, quest'ultimo dalla memoria del disco alla memoria di lavoro del microprocessore, il quale, provvederà ad eseguire il programma. Ecco che in questo modo il sistema operativo lancerà l'esecuzione del programma da noi precedentemente scritto, memorizzato e compilato.

Capitolo 4 - Esempi

4.1 – Esempi di semplici programmi.

Di seguito fornisco alcuni esempi dello stesso programma realizzato in linguaggi diversi. Si tratta del programma “Hello World”, il programma che tutti i manuali di un linguaggio di programmazione propongono come primo esercizio. Infine un esempio di algoritmo in pseudocodice, più precisamente l’algoritmo Bubble Sort per l’ordinamento di un vettore di numeri in ordine crescente o decrescente.

4.2 BASIC

```
10 PRINT "Hello World!"
```

4.3 PASCAL

```
program HelloWorld;  
  
begin  
  writeln('Hello World');  
end.
```

4.4 C++

```
#include <iostream>
```

```
int main()
```

```
{
```

```
    std::cout << "Hello, world!\n";
```

```
}
```

4.5 ASP

```
<%  
    HelloWorldLabel.Text = "Hello, world!";  
%>  
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML  
1.0 Transitional//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-  
transitional.dtd">  
<html xmlns="http://www.w3.org/1999/xhtml" >  
<head runat="server">  
    <title>Untitled Page</title>  
</head>  
<body>  
    <form id="form1" runat="server">  
        <div>  
            <asp:Label runat="server"  
id="HelloWorldLabel"></asp:Label>  
        </div>  
    </form>  
</body>  
</html>
```

4.6 PHP

```
<html>  
<head>  
  <title>Test PHP</title>  
</head>  
<body>  
<?php echo "Hello World!<p>"; ?>  
</body>  
</html>
```

4.7 – JSP

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class HelloWorld extends HttpServlet {

    public void doGet(HttpServletRequest req,
        HttpServletResponse res)
        throws ServletException,
        IOException {

        res.setContentType("text/html");
        PrintWriter out = res.getWriter();

        out.println("<HTML>");
        out.println("<HEAD><TITLE>Hello
World</TITLE></HEAD>");
        out.println("<BODY>");
        out.println("<H1>Hello World</H1>");
        out.println("Today is: " + (new
java.util.Date().toString()) );
        out.println("</BODY></HTML>");
    } // doGet
} // HelloWorld
```

4.8 – VB.NET

```
Private Sub btnHelloWorld_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnHelloWorld.Click
```

```
    MessageBox.Show("Hello World")
```

```
End Sub
```

4.9 – C#

// A Hello World! program in C#.

```
using System;
```

```
namespace HelloWorld
```

```
{
```

```
    class Hello
```

```
    {
```

```
        static void Main()
```

```
        {
```

```
            Console.WriteLine("Hello World!");
```

```
            // Keep the console window open in debug  
mode.
```

```
            Console.WriteLine("Press any key to exit.");
```

```
            Console.ReadKey();
```

```
        }
```

```
    }
```

```
}
```

4.10 – ASP.NET

```
<%  
    HelloWorldLabel.Text = "Hello, world!";  
%>  
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML  
1.0 Transitional//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-  
transitional.dtd">  
<html xmlns="http://www.w3.org/1999/xhtml" >  
<head runat="server">  
    <title>Untitled Page</title>  
</head>  
<body>  
    <form id="form1" runat="server">  
        <div>  
            <asp:Label runat="server"  
id="HelloWorldLabel"></asp:Label>  
        </div>  
    </form>  
</body>  
</html>
```

4.11 – Algoritmo di ordinamento (Bubble Sort).

```
procedure BubbleSort(A:lista di elementi da ordinare)
  scambio ← true
  n ← length(A) - 1
  while (scambio) do
    scambio ← false
    for i ← 0 to n - 1 do
      if (A[i] > A[i + 1]) then //sostituire '>' con '<'
        per ottenere un ordinamento decrescente
          swap ( A[i], A[i+1] )
        scambio ← true
    n ← n-1 //ad ogni passaggio si
  accorcia di uno il ciclo di for
```