

# CONTENUTI E METODOLOGIE PER LA DIDATTICA DELL'INFORMATICA NELLA SCUOLA SUPERIORE

Agostino Lorenzi  
Didamatica 2007

## Abstract

*I programmi degli indirizzi informatici nella scuola superiore risalgono all'inizio degli anni '90. Da allora l'adeguamento allo sviluppo tecnologico è stato fatto dai singoli docenti o dalle singole scuole. Nello stesso tempo è cambiato il profilo professionale per le figure informatiche di primo livello (programmatore) e sono cambiate le richieste di personale da parte delle aziende. L'analisi qui svolta presenta alcune proposte operative per costruire un percorso didattico e definire le metodologie didattiche efficaci, con attenzione anche all'Open Source e alle opportunità offerte dall'e-learning.*

## Introduzione

Nel 1991 il Ministero della Pubblica Istruzione avviò due nuove sperimentazioni assistite per gli indirizzi informatici nella scuola superiore, denominati Mercurio per l'indirizzo ragionieri-programmatori negli Istituti Tecnici Commerciali e Abacus per l'indirizzo periti in informatica negli Istituti Tecnici Industriali [1]. Le sperimentazioni, consolidate dopo il 1995, rappresentarono un punto importante di svolta, perché rinnovarono profondamente il quadro orario e i programmi e recepirono i cambiamenti che andavano emergendo, con la richiesta da parte del mercato del lavoro di una figura professionale di primo livello più aperta alle nuove applicazioni informatiche.

Dal 1991 ad oggi ci sono stati però anni di grandi mutamenti nelle tecnologie e nell'informatica in particolare: si pensi al grande sviluppo di Internet e del Web a partire dalla metà degli anni '90, alla disponibilità di piattaforme hardware di grandi prestazioni, alla possibilità di utilizzare ambienti per lo sviluppo rapido del software.

La contrazione della spesa per la scuola e il travaglio della riforma della scuola hanno di fatto lasciato fermi programmi di studio e aggiornamento dei docenti a 15 anni fa: i cambiamenti sono stati recepiti all'interno della didattica dell'informatica grazie all'aggiornamento continuo dei singoli docenti e alla revisione dei programmi operata a livello di singola scuola. La discussione sul cambiamento dei programmi era stata recentemente avviata con le proposte di OSA (Obiettivi Specifici d'Apprendimento) in occasione del varo della riforma della scuola superiore [2].

In assenza di mutamenti a livello centrale, sono state fatte alcune scelte da parte dei docenti con due riferimenti molto concreti: la disponibilità di libri di testo rinnovati da parte delle case editrici e la tipologia del tema di informatica nella seconda prova dell'esame di Stato.

Il tipo di prova che negli ultimi anni è stata assegnata agli esami di maturità richiede allo studente di sviluppare l'analisi di un problema la cui soluzione

informatica presuppone l'uso delle basi di dati e l'accesso ad esse tramite le tecnologie Web.

Queste sembrerebbero le competenze specifiche richieste allo studente come punto di arrivo, tacitamente condiviso e mai codificato, per il percorso triennale dell'informatica nella scuola superiore.

## Il profilo professionale

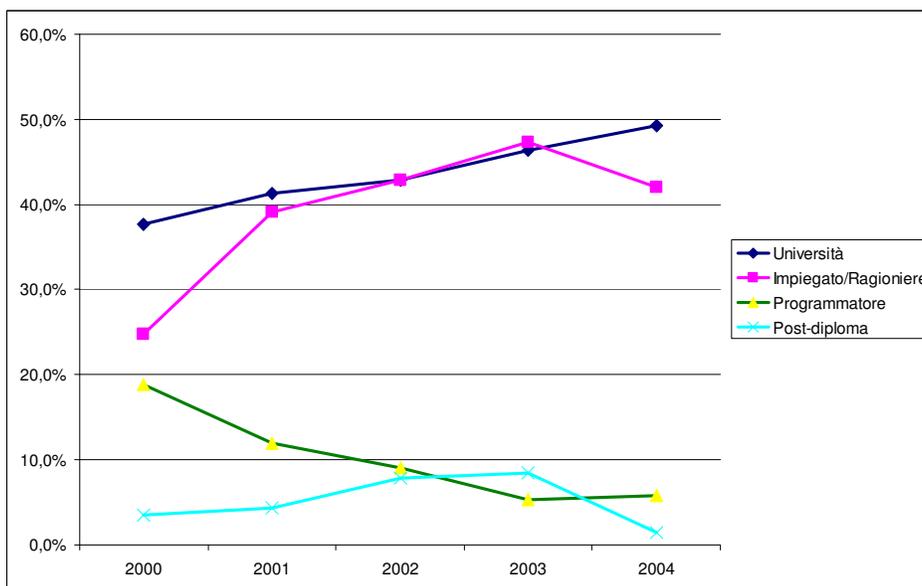
Nella visione delle sperimentazioni Mercurio e Abacus, la figura del programmatore, intesa come esperto di un particolare linguaggio di programmazione e capace di tradurre con precisione gli algoritmi, andava già evolvendosi in quella di esperto di informatica, in grado di recepire gli aspetti fondamentali del sistema informativo aziendale e di utilizzare strumenti diversi per lo sviluppo del software funzionale alle soluzioni informatiche. Quindi non soltanto capacità di produzione e testing del software, ma anche capacità di analisi e progettazione.

In particolare il profilo di ragioniere-programmatore, o meglio di ragioniere esperto in informatica, ha assunto nel tempo una grande rilevanza ai fini dell'occupazione post-diploma, perché è una figura in grado di utilizzare e sviluppare competenze informatiche, sostenute dalle conoscenze derivanti dall'economia aziendale e dal diritto. Questo risponde alle esigenze delle Aziende e degli Enti pubblici, in particolare del ricco tessuto di PMI dell'economia italiana.

Considerando l'andamento degli ultimi anni, si può osservare che l'ingresso nel mondo del lavoro dei neo-diplomati come programmatori è diminuita, mentre è aumentata la percentuale degli studenti che proseguono gli studi all'Università.

Da questo punto di vista è interessante analizzare i risultati di un'indagine svolta tra i diplomati di un Istituto Tecnico Commerciale con indirizzo Mercurio dal 2000 al 2004 e illustrati nel grafico seguente [3].

Fig. 1 – Andamento 2000-2004 sugli sbocchi post-diploma



Il numero delle iscrizioni all'università si è incrementato con l'avvio della riforma dei percorsi universitari (3+2), che ha provocato di fatto lo spostamento in avanti del termine della formazione, almeno alla laurea triennale.

La diminuzione di programmatori neo-diplomati potrebbe essere spiegata anche con il fatto che gli studenti migliori proseguono gli studi, preferibilmente nelle Facoltà di Informatica e Ingegneria Informatica.

Quindi il mercato privilegia l'assunzione di ragionieri-programmatori subito dopo il diploma, con maggiore accento sulla voce "ragioniere", anche se il lato "programmatore" può significare per l'azienda (normalmente una piccola azienda o uno studio professionale) l'opportunità di disporre di una persona in grado di risolvere i più comuni problemi hardware e software e, comunque, di interagire con un eventuale servizio di assistenza tecnica esterna.

A questo si deve aggiungere che, come risulta dalle indagini Excelsior di Unioncamere, le Aziende manifestano ampiamente la richiesta di ulteriore formazione [4].

La tabella seguente presenta a questo proposito dati molto significativi.

Tav. 2 - Assunzioni previste dalle imprese per il 2006 per necessità di ulteriore formazione e altre caratteristiche, secondo l'indirizzo di studio di livello secondario e post-secondario.

|                                      | Totale assunzioni | di cui con necessità di formazione: |                       |                       | di cui:        |                   | di cui: (genere ritenuto preferibile rispetto al titolo segnalato) |             |                   |
|--------------------------------------|-------------------|-------------------------------------|-----------------------|-----------------------|----------------|-------------------|--|-------------|-------------------|
|                                      |                   | Totale                              | con corsi esterni (1) | con corsi interni (1) | fino a 29 anni | età non rilevante | Uomini   | Donne       | Ugualmente adatti |
| <b>TOTALE</b>                        | <b>695.770</b>    | <b>71,3</b>                         | <b>9,3</b>            | <b>20,3</b>           | <b>39,5</b>    | <b>36,8</b>       | <b>43,8</b>  | <b>19,5</b> | <b>36,7</b>       |
| Indirizzo informatico                | 6.730             | 80,9                                | 14,3                  | 32,5                  | 52,8           | 22,7              | 39,1   | 4           | 56,9              |
| Indirizzo amministrativo-commerciale | 76.380            | 76,3                                | 9                     | 23,9                  | 51,2           | 24                | 18,1   | 35,1        | 46,8              |

Questa necessità di ulteriore formazione dopo il diploma è stata soddisfatta, negli anni recenti, in modo positivo dai corsi IFTS (Istruzione e Formazione Tecnica Superiore).

### Linee guida per la didattica dell'informatica

Il continuo sviluppo delle tecnologie dell'informazione e della comunicazione pone l'esigenza di un aggiornamento permanente dei contenuti e delle metodologie di lavoro, per gli utenti finali dei prodotti informatici, ma soprattutto per chi vuole diventare programmatore o, in generale, esperto di software.

In particolare occorre fissare una solida base di partenza sui fondamenti teorici dell'informatica, sulle caratteristiche e le modalità operative dell'ambiente di lavoro con il computer e sui nuclei concettuali di riferimento per la progettazione e lo sviluppo di software applicativo.

Nel corso degli anni sono cambiati in modo significativo, insieme alle prestazioni delle attrezzature hardware, gli ambienti di sviluppo software, gli

ambiti di utilizzo dell'informatica e le modalità di interazione tra l'utente e il computer. In particolare le applicazioni dell'informatica, oltre a consolidarsi nei settori tradizionali dell'automazione industriale e della gestione aziendale, si sono strettamente legate all'uso delle reti, di Internet e della multimedialità.

Si deve allora evitare di fissare l'attenzione su singoli prodotti commerciali o su specifici modelli hardware, che spesso diventano obsoleti in tempi brevi, per cogliere invece, all'interno di essi, gli aspetti che permangono e che devono costituire la base per la costruzione di un profilo professionale:

- l'importanza del lavoro ordinato nella metodologia e nella documentazione,
- la definizione dei modelli di dati,
- la costruzione di interfacce per l'utente nell'accesso ai dati,
- la condivisione di risorse e servizi tramite le reti [2].

## **Contenuti**

La programmazione per il triennio ad indirizzo informatico si basa su alcuni prerequisiti che possono essere completati nell'arco del primo biennio: essi riguardano essenzialmente l'informatica di base, compresa un'introduzione al foglio elettronico e l'uso di strumenti per il Web, quali browser e posta elettronica. In sostanza i contenuti del Syllabus della certificazione ECDL [5], con eventuale esclusione dei moduli sul foglio elettronico e sui database i cui esami possono essere convenientemente sostenuti nelle classi successive.

I temi descritti di seguito possono essere articolati nell'arco dei tre anni secondo la programmazione meglio rispondente alle scelte didattiche del docente e alla situazione della specifica classe.

La parte iniziale del percorso triennale (grosso modo la classe terza) riguarda: risorse hardware e software, sistemi operativi DOS e Windows, progettazione degli algoritmi, strutture della programmazione, strutture di dati (compresi i file sequenziali e random).

I contenuti successivi possono essere raggruppati attorno a tre grandi temi:

### *a) i linguaggi di programmazione*

Il punto più controverso, sul quale esiste oggi una grande varietà di scelte tra i docenti dei diversi Istituti, riguarda i linguaggi di programmazione e gli ambienti di sviluppo software.

Nell'indirizzo Abacus prevalgono tradizionalmente i linguaggi C, C++, Java e, più recentemente, C#, mentre nell'indirizzo Mercurio sono più diffusi i linguaggi Pascal e Visual Basic, anche se la divisione non è così netta.

In particolare il linguaggio Pascal si è consolidato negli anni '70 e '80 come ambiente adatto all'apprendimento di una modalità di progettazione e di programmazione ordinata e rigorosa, in modo da acquisire le competenze di base che consentissero di passare facilmente ad altri ambienti di programmazione.

Più recentemente è stato introdotto ampiamente l'uso del linguaggio Visual Basic e in misura minore il linguaggio Delphi. La programmazione visuale, che è diventata sicuramente un approccio standard nella produzione di software, trova oggi vaste applicazioni in numerosi ambienti di sviluppo software. Un programma non è fatto soltanto di dati e di istruzioni, ma anche di interfaccia per l'utente, che deve essere progettata con attenzione perché sia efficace e amichevole. L'uso degli oggetti grafici, i concetti di proprietà, eventi, metodi,

codice associato agli eventi, sono inoltre di uso comune in tutti i tools di sviluppo software con interfaccia grafica.

*b) modellazione dati, progettazione dei database e accesso ai dati in rete*

Fondamentalmente i contenuti da sviluppare riguardano la modellazione dei dati, la progettazione di basi di dati e l'uso del linguaggio SQL, che è il linguaggio standard per gli ambienti software di gestione dei database.

Per la parte pratica, in una prima fase, si possono implementare database in locale usando il programma Access e il linguaggio SQL, prevedendo anche la presentazione delle caratteristiche dell'ambiente di programmazione in Access per costruire applicazioni complete che utilizzano i database.

Successivamente si può passare alla gestione dei database in rete. Le tecnologie Web riguardano la realizzazione di pagine ASP classiche con ADO, pagine ASP.NET con ADO.NET e pagine Php per l'accesso ai database MySQL. In questo modo, lo studente ha l'occasione di apprendere due linguaggi di scripting per il Web, ASP e Php, oltre a due ambienti di gestione dei database, non solo Access, ma anche MySQL, che è un prodotto Open Source molto utilizzato sui server Internet.

In alternativa si può pensare all'introduzione di SQLServer, anche se quest'ultimo appare adatto piuttosto a un corso post-diploma.

*c) sistemi operativi e reti*

L'argomento dei sistemi operativi deve essere svolto sicuramente dal punto di vista teorico, anche per offrire agli studenti l'occasione per sistematizzare a livello concettuale le esperienze di uso del computer svolte negli anni precedenti. I concetti possono essere poi ripresi, in pratica, con la presentazione del sistema operativo Linux, come secondo (o terzo) sistema operativo dopo Windows (e DOS): di esso sono importanti le caratteristiche dell'interfaccia grafica nelle distribuzioni più recenti, le modalità operative per usare la linea dei comandi e per costruire script nella shell.

Linux è diventato nel corso degli anni un sistema operativo standard per la gestione delle reti e dei server Internet. Dal punto di vista didattico, esso risulta particolarmente adatto per comprendere l'applicazione pratica dei concetti che stanno alla base della teoria dei sistemi operativi.

In particolare risulta molto efficace, soprattutto in fase di addestramento, l'uso di una distribuzione Linux di tipo Live su CD o DVD. Infatti lo studente può in questo modo provare i comandi, avviare i servizi di sistema, provare esempi ed eseguire esercizi senza la necessità di installare il sistema operativo sul proprio computer.

Anche i contenuti riguardanti le tecnologie e i servizi delle reti e di Internet necessitano di una trattazione teorica sulle architetture e i protocolli delle reti moderne e di Internet, insieme alle nozioni di base per la configurazione e l'amministrazione di una rete con Windows e con Linux e per l'uso dei servizi di Internet.

Dal punto di vista pratico è importante fornire gli strumenti per costruire pagine Web, in particolare con i linguaggi HTML e XHTML, i fogli di stile CSS e il linguaggio Javascript, sottolineando anche l'importanza di progettare le pagine in modo che rispettino i requisiti standard di accessibilità e usabilità.

Risulta infine indispensabile introdurre le caratteristiche e le applicazioni del linguaggio XML, considerato il grande sviluppo di questo linguaggio negli anni recenti.

Nella formazione culturale degli studenti devono essere anche presenti le conoscenze relative alle applicazioni dell'informatica e delle reti come supporto al business aziendale, insieme ai meccanismi per la sicurezza e la protezione dei dati. In questo ambito vanno presentate le problematiche relative all'uso delle tecnologie informatiche e di rete per rendere efficiente la Pubblica Amministrazione, secondo le norme e i codici stabiliti negli anni recenti. Può essere opportuno completare questa parte con le nozioni sul diritto dell'informatica, attraverso una panoramica delle norme che riguardano le nuove tecnologie (sicurezza, transazioni commerciali, privacy, reati informatici, validità dei documenti elettronici).

### **Metodologie di lavoro**

L'acquisizione di contenuti informatici aggiornati va di pari passo con l'importanza di una corretta metodologia di lavoro da proporre agli studenti in modo da sviluppare le capacità progettuali che sono insite nella visione moderna del profilo professionale di un tecnico diplomato ed esperto di informatica.

Vengono presentati di seguito due esempi.

#### *a) metodologia di sviluppo di un problema nella programmazione*

L'elemento nuovo della moderna programmazione, a differenza dei linguaggi di programmazione tradizionali, è la presenza dell'interfaccia come componente fondamentale del software. Questo è dovuto a molti motivi, ma soprattutto a tre fondamentali:

- l'uso del computer non è più riservato a specialisti, ma è diffuso a livello di massa;
- l'interazione con l'utente è sempre più amichevole e basata sull'uso di oggetti grafici;
- le tecnologie software hanno avuto una forte accelerazione con lo sviluppo delle applicazioni per il Web.

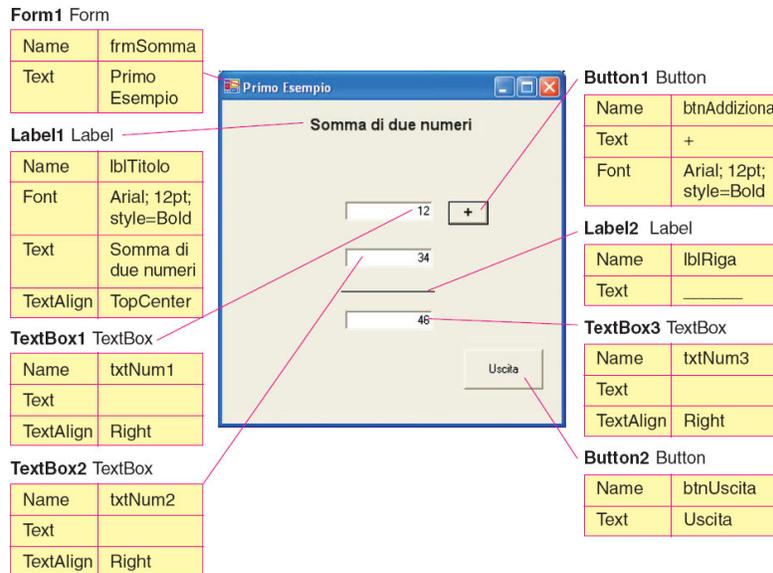
Si aggiunga inoltre che l'uso esteso delle interfacce grafiche introduce nuovi aspetti formativi nell'attività didattica: gli studenti non devono costruire programmi che sono soltanto la traduzione di un procedimento risolutivo, ma devono fare attenzione a tutto quello che può succedere durante la fase di esecuzione del programma, prevedendo gli eventi generati dall'utente, oltre che considerare con attenzione il problema della comunicazione e delle modalità di interazione con l'utente del programma [6].

Le attività dello studente, a partire dal testo del problema alla scrittura del codice sorgente, sono sintetizzate con questo schema:

- analisi del problema con individuazione dei dati di input, di output e una descrizione discorsiva del procedimento risolutivo
- progettazione dell'interfaccia utente con la definizione degli oggetti e del layout (Fig. 3)
- formalizzazione del procedimento in un algoritmo scritto in un linguaggio di pseudocodifica (che utilizza parole del linguaggio naturale) e con un'eventuale rappresentazione grafica mediante diagramma a blocchi

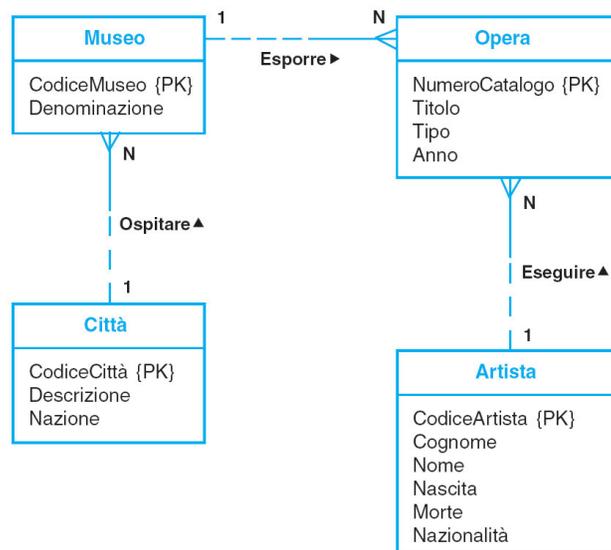
- codifica nel linguaggio di programmazione
- testing e debugging del programma [7].

Fig.3 – Progettazione dell'interfaccia utente



*b) metodologia di modellazione dei dati e implementazione di una base di dati*  
 La progettazione e l'implementazione di una base di dati deve essere preceduta da un'attenta attività di modellazione dei dati: le fasi successive infatti possono creare le strutture logiche dei dati e la realizzazione della basi di dati attraverso semplici regole di derivazione.

Fig. 4 – Modello E/R



- L'attività che lo studente deve svolgere è formata quindi dai seguenti punti:
- Analisi del problema, definizione delle entità, attributi e associazioni
  - Modello concettuale (E/R) (come nell'esempio della Fig. 4)

- Validazione del modello E/R con le regole di lettura
- Derivazione delle tabelle dal modello concettuale
- Schema di descrizione dei dati con specificazione di tabelle, colonne, chiavi primarie ed esterne, formato e dimensione dei campi
- Funzionigramma con la schema grafico dell'albero delle procedure e delle sottoprocedure
- Schema delle risorse di sistema con indicazione dei flussi di input e di output
- Interrogazioni in pseudocodifica (come nell'esempio della Fig. 5)
- Codifica nel linguaggio SQL [8].

Fig.5 – Interrogazioni in pseudocodifica

*Elenco degli iscritti (cognome, nome, tipo socio) di una Nazione*

|   |  |
|---|--|
| 1. $\sigma_P$ Nazioni                                       | Temp1 = Selezione di Nazioni per <b>DescrNazione = [nazione scelta]</b>                  |
| 2. Temp1 <sub>PK</sub> $\bowtie$ Iscritti <sub>FK</sub>     | Temp2 = Congiunzione di Temp1 su <b>ChiaveNazione</b> e Iscritti su <b>CodiceNazione</b> |
| 3. Temp2 <sub>FK</sub> $\bowtie$ TipiAdesione <sub>PK</sub> | Temp3 = Congiunzione di Temp2 su <b>CodiceTipo</b> e TipiAdesione su <b>ChiaveTipo</b>   |
| 4. $\Pi_L$ Temp3  | Proiezione di Temp3 su <b>Cognome, Nome, DescrTipo</b>                                   |

L'uso del linguaggio di pseudocodifica consente di rappresentare con precisione le operazioni dell'algebra relazionale, evidenziando anche i problemi di ottimizzazione delle interrogazioni [9].

## Open Source

Anche nella scuola, seppur lentamente, si sta diffondendo tra studenti e docenti la cultura dell'Open Source e dell'uso di software con licenza GNU [10].

L'ampio sviluppo di progetti software di questo tipo, disponibili in Internet, lascia prevedere la possibilità di costruire percorsi didattici di informatica basati esclusivamente su prodotti Open Source.

Questa categoria comprende in primo luogo il sistema operativo Linux, con le sue applicazioni e i suoi servizi. Poi si deve considerare la suite OpenOffice.org ([www.openoffice.org](http://www.openoffice.org)) che, nelle sue versioni più recenti, rende disponibili funzionalità del tutto simili ai prodotti Office di Microsoft. In particolare la release 2.1 ha migliorato notevolmente il programma OpenOffice.org Base per la gestione dei database.

Per i database un ruolo essenziale è sicuramente svolto dal prodotto MySQL, che si accompagna tradizionalmente all'uso del linguaggio Php per la realizzazione di applicazioni Web. Per gestire i database MySQL, oltre a phpMyAdmin, sono ora disponibili dal sito [www.mysql.com](http://www.mysql.com) due strumenti software di grande interesse che facilitano le operazioni attraverso l'interfaccia grafica: MySQL Query Browser e MySQL Administrator.

Più complesso e variegato è il discorso sui linguaggi di programmazione. Da anni sono disponibili ambienti di programmazione con licenza libera: linguaggi Pascal, C e C++, per esempio le versioni Dev-C++ e Dev-Pascal di [www.bloodshed.net](http://www.bloodshed.net), il linguaggio Java ([java.sun.com](http://java.sun.com)) con gli ambienti IDE NetBeans ([www.netbeans.org](http://www.netbeans.org)) e BlueJ ([www.bluej.org](http://www.bluej.org))

Risorse per lo sviluppo rapido del software, più recenti e di grande interesse, sono il linguaggio Python ([www.python.org](http://www.python.org)) e gli ambienti Mono ([www.mono-project.com](http://www.mono-project.com)) ed Eclipse ([www.eclipse.org](http://www.eclipse.org)). Un progetto avviato negli ultimi anni, che potrebbe offrire un'alternativa al Visual Basic, è rappresentato da GambaS ([gambas.sourceforge.net](http://gambas.sourceforge.net)).

Occorre segnalare, per completezza, che lo scorso anno Microsoft ha rilasciato le versioni Express dei suoi ambienti di programmazione in forma libera per studenti: Visual Studio (Visual Basic, C#, J#, C++) oltre a Visual Web Developer e SQLServer2005 ([www.microsoft.com/vstudio/express](http://www.microsoft.com/vstudio/express)). Inoltre per molti dei prodotti software Open Source citati in precedenza è disponibile anche la versione per il sistema operativo Windows.

## **E-learning**

Un indirizzo informatico non può prescindere dalle opportunità offerte dalle tecnologie Web per quanto riguarda l'apprendimento attivo e collaborativo in rete (e-learning).

Tra l'altro la crescente disponibilità di prodotti di tipo open source offre la possibilità di affidare l'installazione e la gestione della piattaforma e-learning agli studenti delle classi finali, i quali in questo modo possono realizzare un'applicazione pratica dell'uso di database in rete e del linguaggio di scripting per la personalizzazione del codice open.

L'e-learning nella scuola nasce dall'esigenza di introdurre le nuove tecnologie per migliorare le modalità di insegnamento e i processi di apprendimento, di motivare gli studenti e nello stesso tempo di favorire la riprogettazione dei percorsi didattici dei docenti.

L'e-learning è visto soprattutto come supporto alla didattica in presenza e non come sostitutivo dell'attività in aula e in laboratorio. Esso consente l'accesso dello studente secondo tempi e ritmi propri, in particolare per le attività di recupero e rinforzo. I risultati attesi riguardano la possibilità di personalizzazione dell'insegnamento e la riduzione della distanza docente-allievo [11].

Le attività sono principalmente di tipo asincrono, anche se non si possono escludere attività sincrone, quali chat e condivisione di applicazioni.

Si deve sottolineare che l'e-learning per lo studente non è solo apprendimento disciplinare, ma è anche acquisizione di una metodologia di lavoro e di una mentalità collaborative.

Si deve inoltre combattere l'idea che fare e-learning corrisponda a un carico di lavoro aggiuntivo per lo studente: l'uso della didattica in rete può anzi consentire risparmi di tempo e anche la possibilità di trattare argomenti disciplinari in modo più efficace rispetto alla tradizionale attività in presenza.

Il docente, da parte sua, può rilevare, dall'analisi dei lavori individuali, i punti di forza e di debolezza degli studenti rispetto alle conoscenze e abilità previste dal programma di studio, ricavando informazioni utili per la programmazione e la gestione didattica dell'intera classe.

Esempi di utilizzo dell'e-learning per il supporto alla didattica in presenza sono:

- Percorsi di recupero
- Percorsi di potenziamento e approfondimento
- Correzione compiti a casa

- Correzione prove individuali svolte in classe
- Repository per le attività di laboratorio (Quaderno elettronico di laboratorio) [12].

### **Bibliografia**

- [1] Ministero della Pubblica Istruzione, Direzione Generale Istruzione Tecnica, 1995, *Progetto Mercurio, Proposta di ristrutturazione del corso per Ragionieri e Periti Programmatori*, MPI
- [2] Casadei G., Colazzo L., De Lotto I., Fadini B. (coordinatore), Martini S., Mauri G., Mentrasti P., Negri M., Schaerf M., Visaggio G., 2003, *Informatica nei Licei nel contesto della riforma della scuola*, Relazione finale di un gruppo di lavoro AICA ad uso delle commissioni del MIUR sulla riforma dei cicli scolastici, Documenti di Mondo Digitale, AICA, supplemento al n. 4, dicembre 2003.
- [3] Istituto "B. Belotti", Bergamo, 2000-2004, *Autoanalisi, Indagine sui diplomati*, [www.istitutobelotti.it/autoanalisi](http://www.istitutobelotti.it/autoanalisi)
- [4] Unioncamere, Ministero del Lavoro, Sistema Informativo Excelsior per l'occupazione e la formazione, 2006, *I principali risultati 2006, Livelli formativi e indirizzi di studio*, <http://excelsior.unioncamere.net/hyper/ver4/index.htm>
- [5] AICA, 2003, *Syllabus ECDL (European Computer Driving Licence) Core 4.0*, [www.ecdl.it](http://www.ecdl.it)
- [6] Lorenzi A., 2003, *Pascal: kiss it goodbye?*, Didamatica, Genova
- [7] Lorenzi A., Rossi D., 2004, *Informatica: Teoria e programmazione in Visual Basic*, Edizioni Atlas, Bergamo
- [8] Lorenzi A., Cavalli E., 2005, *Informatica: Le basi di dati e il linguaggio SQL. Access, MySQL, Database in rete*, Edizioni Atlas, Bergamo
- [9] Tanca L., 2006, *Linguaggi Formali per Basi di Dati*, Mondo digitale, anno V, n. 4, dicembre 2006
- [10] MIUR, Direzione Generale per i Sistemi Informativi, 2004, *Indagine sulle risorse tecnologiche per la didattica nella scuola italiana*
- [11] Di Blasi G., Lorenzi A., 2005, *L'e-learning nel supporto alla didattica in presenza: motivazioni per l'apprendimento e riprogettazione dei percorsi didattici*, Expo e-learning, Ferrara
- [12] Lorenzi A., 2006, *E-learning: risultati attesi e utilizzo nella didattica di tutti i giorni*, Convegno "Elearning e nuove metodologie per l'insegnamento-apprendimento", Bergamo

Sito Web personale: [www.lorenzi.info](http://www.lorenzi.info)