

Un curriculum per separare l'informatica dall'applicativa* in un Liceo scientifico opzione scienze applicate

Gionata Massi

IIS Savoia Benincasa, Ancona (AN)

Sommario

Questo articolo presenta un curriculum per l'insegnamento dell'informatica nel Liceo scientifico opzione scienze applicate che ambisce di introdurre la materia come disciplina scientifico/ingegneristica¹.

Il curriculum è orientato all'apprendimento dei fondamenti epistemologici dell'informatica in accordo con i traguardi di competenza fissati nelle Indicazioni nazionali però presenta una selezione di argomenti e contenuti che sconcerta le attese di studenti, genitori e colleghi abituati ad un percorso di addestramento all'uso delle tecnologie informatiche.

La didattica di alcuni argomenti richiede l'uso del PC e del software², come implicitamente richiesto dalla Indicazioni nazionali, senza essere in contraddizione con gli obiettivi epistemologici.

Il curriculum è presentato come un elenco di traguardi di competenza, espressi in termini di abilità verificabili nel primo biennio, nel secondo biennio e nel quinto anno, raggruppati come in argomenti e aree tematiche³.

La sperimentazione del curriculum è stata effettuata tra gli anni scolastici 2018/19 e 2022/23 coinvolgendo due sezioni e, a causa della sopraggiunta emergenza sanitaria, non è stato possibile condurre analisi comparative sul raggiungimento degli obiettivi di competenza rispetto alle classi parallele che non seguivano lo stesso percorso didattico.

In conclusione il curriculum qui proposto può essere utilizzato come un repertorio di traguardi di apprendimento al quale gli insegnanti possono attingere liberamente per la progettazione e la programmazione didattica dell'informatica nel proprio istituto; esso è un contributo verso l'insegnamento della materia intesa come disciplina scientifica e tecnica piuttosto che come applicativa e può essere meritevole di ulteriore sperimentazione. Il limite di applicazione di questo percorso è l'assenza di materiali di studio di elevata qualità in lingua italiana.

Indice

1	Introduzione	2
2	Curricolo	3
2.1	Primo biennio	3
2.2	Secondo biennio	6
2.3	Quinto anno	7
3	Sperimentazione	8

*Neologismo trovato in alcuni lucidi del prof. Mattia Monga per indicare l'uso del computer o di applicativi specifici

¹Harold Abelson nella prima lezione del corso MIT 6.001 Structure and Interpretation of Computer Programs sembra essere in disaccordo sulla natura scientifica dell'informatica

²Nella scelta del software ci si è limitati alle sole applicazioni rilasciate come software libero o, laddove non fosse possibile, a sorgente aperto.

³Si vedano le Indicazioni nazionali [9]

4 Conclusioni

9

1 Introduzione

Gli insegnanti di informatica dei licei italiani non dispongono di estesi sillabi ministeriali, programmi nazionali o dettagliati repertori di competenze disciplinari che guidino l'azione didattica in modo uniforme sul territorio nazionale. Essi devono progettare e programmare l'attività didattica, per ogni scuola e classe, cercando un compromesso fra vincoli normativi nazionali, quadri europei⁴ e aspettative di tanti attori.

L'attività di progettazione e programmazione didattica si realizza come se si dovesse ottimizzare euristicamente un modello di programmazione matematica multiobiettivo nel quale le variabili decisionali sono gli argomenti e i contenuti utili a sviluppare le competenze proprie della materia scolastica, i vincoli sono dati dalle Indicazioni nazionali [9] e dal Piano Triennale dell'Offerta Formativa della scuola, e gli obiettivi sono tanti e in contrasto tra loro: venire incontro alle richieste del territorio, adeguarsi alle aspettative di studenti, genitori e altri insegnanti, trovare argomenti e contenuti presenti nel libro di testo in adozione, laddove questo ci sia, riutilizzare il materiale didattico già disponibile, incentivare gli studenti al conseguimento di certificati⁵ e non addentrarsi in saperi che gli insegnati non possiedono⁶.

Nell'esperienza dell'autore, l'aspettativa dell'utenza e del corpo docente si concentra verso un percorso di applicativa* volto a fornire agli studenti le abilità d'uso di qualche programma applicativo con il quale interagire tramite un'interfaccia grafica che esibisce un comportamento *What You See Is What You Get*⁷ (WYSIWYG). L'insieme dei programmi applicativi richiesti dagli insegnanti di altre materie include sempre un certo programma per l'elaborazione del testo e un altro per la predisposizione dei lucidi di supporto alla presentazione. A seconda dei consigli di classe si possono unire all'insieme dei software i fogli elettronici e alcuni sistemi per la rappresentazione geometrica e/o per il calcolo numerico o simbolico.

I libri di testo del biennio sono spesso adatti a soddisfare questo genere di aspettative, anche per rispettare le Indicazioni nazionali, e solo negli ultimi tre anni oltre all'applicativa introducono un linguaggio di programmazione⁸, spesso uno solo e frequentemente il C++⁹.

La situazione nelle nazioni anglofone, le uniche per le quali l'autore è riuscito ad effettuare una ricerca sullo stato dell'arte, appare diversa. Vi sono curricula molto dettagliati che fanno

⁴Agli insegnanti di informatica è richiesto in particolar modo di conformarsi al Digital Competences Framework (DigComp 2.2), con elenca oltre 250 competenze digitali legate alle *tecnologie digitali*. La traduzione del quadro di riferimento in lingua italiana utilizza il termine *digitale* per *digit* come per riferirsi alla pressione delle dita sulla tastiera o, più probabilmente, il falso amico *digit* non si traduce più con il termine *numerico*.

⁵Spesso usare un sistema di valutazione conformità ai *framework di competenze* comunitari, le scuole sono sede d'esame per certificazioni sull'uso di software applicativi

⁶Il requisito di accesso all'insegnamento per la classe di concorso *A-41 - Scienze e tecnologie informatiche* è molto più lasco rispetto a quelle di altre sezioni di concorso e molti insegnanti, specie quelli meno giovani, non hanno sostenuto alcun esame di informatica nel percorso universitario.

⁷Questo termine indica che quando usi un'applicazione per modificare un documento, *quello che vedi* nell'interfaccia di modifica è *quello che ottieni* nell'interfaccia di visualizzazione. L'applicazione di modifica e quella di visualizzazione possono coincidere.

⁸Nel primo biennio i linguaggi di programmazione sono sostituiti da ambienti grafici per la programmazione visuale a blocchi e il termine *programmazione* è regredisce in favore di *coding*

⁹Bjarne Stroustrup sembra sorpreso dal fatto che il C++ sia ampiamente utilizzato nella didattica in quanto tale linguaggio non è il più compatto e pulito e nella genesi del linguaggio non vi sono elementi sulla natura didattica.

apparire il nostro processo di programmazione didattica come empirico, eterogeneo, orientato all'uso delle applicazioni e non supportato da materiale di studio valido¹⁰.

Negli Stati Uniti si seguono curricula e attività didattiche progettate in dettaglio, come [7] che propone scopi affini alle nostre *indicazioni nazionali* [9]; le esperienze didattiche e i curricula sono spesso raccolte in cataloghi [12]. Le piattaforme di apprendimento online come Code.org [11] e Khan Academy [10] offrono corsi gratuiti basati sui curricula più noti come lo Advanced Placement Computer Science [3]. La comunità scientifica si occupa da tempo della didattica dell'informatica anche tramite lo sviluppo di linguaggi di programmazione¹¹ e ambienti per l'insegnamento¹² ([1] e [2], [6]). Anche in Gran Bretagna c'è molta attenzione all'insegnamento dell'informatica e alla formazione professionale dei docenti¹³ [8]. Di rilievo appaiono anche i progetti didattici della Nuova Zelanda e tra quelli dedicati all'informatica vi è la *Computer Science Field Guide* [5], una guida per insegnanti e studenti che è stata tradotta in varie lingue¹⁴.

Questo documento presenta nel par. 2 la proposta di curriculum che l'autore ha progettato, e seguito con due sezioni, per ridurre i contenuti dell'applitmatica e avvicinare gli studenti alle problematiche, al sapere e ai metodi dell'Informatica. Nel par. 3 illustra la sperimentazione e chiarisce le cause della mancata osservazione comparativa con altre classe del medesimo ciclo e nel par. 4 determina le condizioni dell'applicabilità del curriculum in base alla sperimentazione svolta.

2 Curriculum

Per esigenze di brevità, il curriculum è riportato in modo parziale e si rimanda il lettore alla pagina <https://github.com/gionatamassibenincasa/progettazione-didattica-informatica>.

2.1 Primo biennio

2.1.1 Scrittura tecnico-scientifica al calcolatore

Area tematica: Elaborazione digitale dei documenti (DE)

1. Identificare gli attori di una comunicazione
2. Distinguere tra dati, informazioni e conoscenza
3. Aggiungere l'autore e il titolo nel frontespizio di una relazione
4. Suddividere un testo nei suoi elementi strutturali ed aggiungere i titoletti
5. Scrivere tabelle, immagini e altri contenuti flottanti con didascalie

¹⁰Il sistema scolastico italiano ha molte peculiarità che lo rendono estremamente formativo, o almeno questa è l'impressione che se ne ricava quando si assiste ai colloqui pluridisciplinari degli studenti che hanno frequentato un anno scolastico all'estero, ma questa affermazione sembra non essere generalizzabile per l'informatica.

¹¹Albeson e Stroustrup, per citare due creatori di linguaggi di programmazione, considerano il linguaggio di programmazione come uno strumento per esprimere le idee sotto forma di codice.

¹²Seymour Papert e Mitchel Resnick sono così famosi tra gli insegnanti italiani che è inutile citarli ma il loro contributo ora viene usato per spingere verso il *coding* che pare cosa diversa dalla programmazione.

¹³Un progetto di riferimento è [4] e comunità e fondazioni di produttori di hardware Open Source che propongono curricula completi.

¹⁴La guida è tradotta in tedesco, spagnolo e polacco. Alla lista manca l'italiano e non sono segnalati progetti di traduzione nella lingua italiana.

6. Scrivere con elenchi ordinati, non ordinati e descrittivi
7. Enfatizzare il testo selezionando un carattere tipografico opportuno

L'argomento è introdotto per soddisfare sia il vincolo ministeriale sulla produzione di documenti elettronici. La videoscrittura si colloca di buon diritto nell'applicativa per cui si è scelto l'argomento della comunicazione tecnico/scientifica e un linguaggio di dominio specifico dotato di grammatica libera dal contesto. Lo strumento di elaborazione del testo diventa così un linguaggio avente una grammatica espressa tramite il suo diagramma sintattico e l'attività di videoscrittura diventa un'approccio ai linguaggi formali e alla programmazione.

2.1.2 Dati e codifiche

Area tematica: Architettura dei computer (AC)

1. Convertire un numero da base due a base dieci
2. Convertire un numero da base dieci a base due
3. Codificare un testo usando codifiche binarie dei caratteri
4. Decodificare un testo codificato in binario
5. Comprendere le esigenze che hanno condotto allo sviluppo dello standard UNICODE
6. Codificare un'immagine monocromatica mediante run-length encoding
7. Decodificare un'immagine monocromatica codificata mediante run-length encoding

L'argomento è proposto per soddisfare la richiesta di introdurre lo studente alla codifica binaria senza limitarsi ai codici ASCII e Unicode citati nelle Indicazioni nazionali. Verrà ampiamente ripreso al quinto anno quando, nell'analisi degli algoritmi numerici, sono trattate le codifiche in virgola mobile e gli errori di troncamento.

2.1.3 Problemi, modelli, soluzioni e algoritmi

Area tematica: Algoritmi e linguaggi di programmazione (AL)

1. Saper formalizzare un problema di ricerca
2. Simulare l'esecuzione dell'algoritmo di ricerca lineare
3. Simulare l'esecuzione dell'algoritmo di ricerca binaria
4. Saper formalizzare il concetto di ordinamento di una sequenza
5. Simulare l'algoritmo di ordinamento per selezione, per inserimento e a bolle
6. Calcolare il numero di confronti e di scambi degli algoritmi di ordinamento basati su confronti e scambi
7. Comprendere i criteri di scelta di un algoritmo rispetto ad altri
8. Astrarre il modello di semplici problemi di natura quantitativa e descrivere algebricamente il procedimento di soluzione

9. Simulare l'esecuzione di un programma

Il concetto di algoritmo viene formalizzato usando i concetti di problema, modello, codifica, istanza, soluzione ed esecutore.

2.1.4 Interagire col sistema operativo tramite la shell

Area tematica: Sistemi operativi (SO)

1. Creare, rinominare, spostare e cancellare un file
2. Organizzare i file nelle directory
3. Invocare un programma
4. Elencare file e directory
5. Creare una pipe anonima tra due o più utility
6. Usare comandi per la data, l'estrazione della testa e della coda di un file di testo, l'ordinamento delle linee, l'estrazione di campi

Le Indicazioni nazionali richiedono la conoscenza delle funzionalità dei sistemi operativi più comuni e si è scelto di orientare lo studio al sistema operativo GNU/Linux in quanto esso è il sistema operativo più eseguito¹⁵ e meglio documentato, oltre a soddisfare il requisito di essere software libero.

2.1.5 Documenti elettronici per il web

Area tematica: Elaborazione digitale dei documenti (DE)

1. Riconoscere un linguaggio basato sui marcatori
2. Usare i tag HTML per creare un documento valido
3. Usare i tag semantici per strutturare il contenuto
4. Saper realizzare un semplice sito web statico curando solo il contenuto
5. Collegare un foglio di stile ad una pagina web
6. Modificare lo stile del testo
7. Modificare l'aspetto delle scatole
8. Saper realizzare un semplice sito web statico curando anche la presentazione

I linguaggi specifici di dominio del web per il contenuto e la presentazione sono introdotti nel primo biennio ma vengono esplorati anche nel secondo biennio.

¹⁵Si veda https://en.wikipedia.org/wiki/Usage_share_of_operating_systems

2.2 Secondo biennio

2.2.1 Fogli elettronici

Area tematica: Elaborazione digitale dei documenti (DE)

1. Usare le operazioni di filtraggio, riduzione e mappa nel foglio di calcolo
2. Usare il foglio di calcolo per modellizzare e risolvere le problematiche d'interesse per il corso di studi
3. Usare il risolutore per problemi di scelta
4. Impostare problemi a variabili intere di ammissibilità e di ottimizzazione vincolata con il risolutore
5. Descrivere l'ordine delle operazioni di aggiornamento in un foglio di calcolo

Si propongono le funzionalità del foglio di calcolo che corrispondono a costrutti tipici della

2.2.2 Paradigmi di programmazione

Area tematica: Algoritmi e linguaggi di programmazione (AL)

1. Saper realizzare algoritmi usando le funzioni e i costrutti di sequenza, selezione e iterazione (paradigma strutturato)
2. Riconoscere la ricorsione come caratteristica sintattica
3. Riconoscere se il processo generato da una procedura sintatticamente ricorsiva è lineare o ricorsivo
4. Progettare semplici basi di fatti e regole con paradigma logico
5. Saper realizzare semplici algoritmi mediante funzioni sintatticamente ricorsive
6. Saper attraversare semplici strutture dati definite ricorsivamente

L'autore non è riuscito a decodificare le richieste ministeriali di *implementazione di un linguaggio di programmazione, metodologie di programmazione, sintassi di un linguaggio orientato agli oggetti*, pertanto si è scelto di presentare il paradigma della programmazione strutturata (con funzioni definite nell'ambito di una funzione), quello logico e quello funzionale. Alla programmazione orientata agli oggetti, richiesta dalle Indicazioni, sono dedicati altri moduli.

2.2.3 Applicazioni web

Area tematica: Algoritmi e linguaggi di programmazione (AL)

1. Saper scrivere semplici script interpretabili dal web browser
2. Saper usare le API di accesso al modello di documento di una pagina HTML
3. Strutturare i dati per permettere la costruzione programmatica di una pagina HTML
4. Realizzare una semplice web app interattiva

Questo argomento introdurre alcuni concetti della programmazione orientata agli oggetti quali quello di istanza di classe, interfaccia, metodi e proprietà.

2.2.4 Problemi su grafi e combinatorici

Area tematica: Algoritmi e linguaggi di programmazione (AL)

1. Saper rappresentare un grafo
2. Realizzare gli algoritmi di visita di un albero binario: preordine, postordine, simmetrico
3. Visitare un grafo in profondità e in ampiezza
4. Simulare l'algoritmo di Dijkstra
5. Rappresentare sul foglio di calcolo il problema dello zaino

2.3 Quinto anno

2.3.1 Reti di calcolatori

Area tematica: Reti di computer (RC)

1. Descrivere i componenti hardware e software che costituiscono una rete dati
2. Descrivere le applicazioni di rete quali quelle del www e della posta elettronica
3. Descrivere i livelli di rete del modello ISO/OSI
4. Riconoscere vantaggi e svantaggi di alcune topologie di rete

2.3.2 Internet e servizi

Area tematica: Struttura di Internet e servizi (IS)

1. Descrivere i servizi di rete con particolare riferimento al www
2. Descrivere gli strati del modello TCP/IP
3. Comprendere i principali rischi di sicurezza nelle comunicazioni digitali
4. Valutare l'uso di tecniche per raggiungere determinati livelli di riservatezza, integrità e disponibilità dei dati

2.3.3 Funzioni di ordine superiore

Area tematica: Computazione, calcolo numerico e simulazione (CS)

1. Definire funzioni che hanno come argomento una funzione di variabile reale e un valore reale e che restituiscono un valore reale
2. Definire funzioni che hanno come argomento una funzione di variabile reale e che restituiscono una funzione di variabile reale
3. Descrivere gli algoritmi in termini di applicazione e composizione di funzioni
4. Manipolare gli array usando le tecniche: filter, map, reduce.

2.3.4 Radici di funzioni non lineari

Area tematica: Computazione, calcolo numerico e simulazione (CS)

1. Saper applicare il metodo di Erone di Alessandria per il calcolo delle radici quadrate
2. Riconoscere l'esistenza di una radice di una funzione dato un intervallo
3. Saper effettuare la ricerca numerica di una radice tramite il metodo di bisezione
4. Realizzare in un linguaggio di programmazione e al foglio di calcolo il metodo di bisezione
5. Saper effettuare la ricerca numerica di una radice tramite il metodo delle tangenti (di Newton)
6. Realizzare in un linguaggio di programmazione e al foglio di calcolo il metodo delle tangenti (di Newton)
7. Saper riconoscere l'equivalenza di due metodi iterativi

2.3.5 Ottimizzazione 1-dimensionale

Area tematica: Computazione, calcolo numerico e simulazione (CS)

1. Riconoscere l'esistenza di un massimo e di un minimo
2. Metodo di Newton per l'ottimizzazione
3. Approssimare il punto di minimo locale usando il metodo di Newton per l'ottimizzazione
4. Realizzare in un linguaggio di programmazione e al foglio di calcolo il metodo di Newton per l'ottimizzazione
5. Realizzare in un linguaggio di programmazione e al foglio di calcolo il metodo della sezione aurea

3 Sperimentazione

L'azione didattica ispirata da tale curriculum si è protratta dall'anno scolastico 2018/19, quando due classi prime hanno iniziato il percorso, a quello 2022/23 quando gli stessi studenti hanno conseguito il diploma.

Le misure adottate per fronteggiare l'emergenza sanitaria per COVID-19 hanno condizionato negativamente la sperimentazione e invalidato la rilevazione dei risultati. È stato impossibile condurre un'analisi sul raggiungimento degli obiettivi di competenza per classi parallele, previsto al termine del secondo anno, poi del quarto e del quinto. In particolare, al termine del secondo anno gli studenti non erano nelle aule scolastiche e anche il terzo anno si è svolto quasi interamente a distanza, senza consentire una valutazione affidabile delle competenze. Non si è quindi riusciti a seguire in modo accurato il curriculum e non sono state effettuate le prove con le classi parallele del quarto e quinto anno per via della forte eterogeneità con le altre classi.

La sperimentazione si è conclusa e non è stata riproposta a causa dell'esigenza di uniformare l'offerta didattica tra classi parallele e per la difficoltà di reperire validi supporti didattici.

4 Conclusioni

In conclusione si considera il curriculum qui presentato come una proposta coerente con i traguardi di competenze fissati dal Ministero dell'Istruzione e del Merito e orientata alla introduzione dell'Informatica come disciplina scientifica piuttosto che come apprendistato all'uso delle applicazioni.

Il curriculum può essere una guida per la progettazione e la programmazione didattica dell'informatica alla quale gli insegnanti liceali possono attingere per la costruzione del percorso didattico del proprio istituto ma presenta una difficoltà di realizzazione dovuta alla mancanza di materiali di studio di qualità elevata in lingua italiana, quali potrebbero essere i libri di testo o di Open Educational Resource (OER).

Per superare questa difficoltà servirebbe la costruzione di una comunità professionale che contribuisca ai progetti esistenti, come il repertorio di OER del Politecnico di Torino [13], al miglioramento dei libri di testo e delle piattaforme didattiche, e alla traduzione professionale dei migliori progetti didattici in lingua straniera.

Riferimenti bibliografici

- [1] Harold Abelson e Gerald Jay Sussman. *Structure and Interpretation of Computer Programs*. Versione open access: <https://sarabander.github.io/sicp/>. The MIT Press, 1996, p. 683. ISBN: 9780262510875.
- [2] Harold Abelson et al. *Structure and Interpretation of Computer Programs: JavaScript Edition. JavaScript Edition*. Versione open access: <https://sourceacademy.org/sicpjs/index>. MIT Press, 2022. ISBN: 9780262543231.
- [3] College Board. *AP Computer Science Principles*. 2023. URL: <https://apstudents.collegeboard.org/courses/ap-computer-science-principles>.
- [4] National Centre for Computer Education. *Helping you teach computing*. 2023. URL: <https://teachcomputing.org/>.
- [5] Computer Science Education Research Group at the University of Canterbury, New Zealand. *Computer Science Field Guide*. 2023. URL: <https://www.csfieldguide.org.nz/>.
- [6] Matthias Felleisen et al. *How to Design Programs. An Introduction to Programming and Computing*. Versione Open Access: <https://htdp.org/>. MIT Press, 2018, p. 792. ISBN: 9780262534802.
- [7] Kathi Fisler et al. «Evolving a K-12 Curriculum for Integrating Computer Science into Mathematics». In: *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education*. SIGCSE '21. Virtual Event, USA: Association for Computing Machinery, mar. 2021, pp. 59–65. ISBN: 9781450380621. DOI: [10.1145/3408877.3432546](https://doi.org/10.1145/3408877.3432546). URL: <https://doi.org/10.1145/3408877.3432546>.
- [8] Brian Fowler e Emiliana Vegas. «How England Implemented Its Computer Science Education Program.» In: *Center for Universal Education at The Brookings Institution* (2021).

- [9] Il Ministro dell'Istruzione, dell'Università e della Ricerca. *Schema di regolamento recante 'Indicazioni nazionali riguardanti gli obiettivi specifici di apprendimento concernenti le attività e gli insegnamenti compresi nei piani degli studi previsti per i percorsi liceali di cui all'articolo 10, comma 3, del decreto del Presidente della Repubblica 15 marzo 2010, n. 89, in relazione all'articolo 2, commi 1 e 3, del medesimo regolamento.'* Allegato F. Ministero dell'Istruzione, dell'Università e della Ricerca. Lug. 2010. URL: <https://www.gazzettaufficiale.it/eli/id/2010/12/14/010G0232/sg>.
- [10] Sal Khan. *Khan Academy*. 2023. URL: <https://www.khanacademy.org/>.
- [11] Code.org Team. *Code.org*. 2023. URL: <https://code.org/>.
- [12] CSforALL Team. *Computer Science for ALL Students*. 2023. URL: <https://www.csforall.org/>.
- [13] Politecnico di Torino. *Free Architecture for Remote Education*. 2023. URL: <https://fare.polito.it/>.