

Un curriculum per separare l'informatica dall'applitmica* 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 piuttosto che una formazione professionale sull'uso dei software applicativi, come talvolta accade nelle scuole italiane.

Il curriculum è orientato all'apprendimento dei fondamenti epistemologici dell'informatica in accordo con i traguardi di competenza fissati nelle Indicazioni nazionali ma presenta una selezione di argomenti e contenuti che sconcertano studenti, genitori e colleghi perché non rispondono alle attese di un percorso di addestramento all'uso delle tecnologie informatiche.

La didattica di alcuni argomenti richiede l'uso del PC e del software² e questo è implicitamente richiesto dalla Indicazioni nazionali e non è in contraddizione con gli obiettivi epistemologici.

Il curriculum è presentato come un estratto della progettazione didattica nel primo biennio, nel secondo biennio e nel quinto anno elencando un sottoinsieme di argomenti, la collocazione all'interno dell'area tematica³, i traguardi di competenza espressi come abilità verificabili ed eventuali annotazioni.

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

In conclusione il curriculum qui proposto può essere utilizzato come un catalogo di traguardi di apprendimento dal quale gli insegnanti possono attingere liberamente per la progettazione e la programmazione didattica dell'informatica del proprio istituto; esso è un contributo verso l'insegnamento dell'informatica come disciplina scientifica e tecnica piuttosto che come un'addestramento all'uso di strumenti tecnologici e può essere meritevole di ulteriore sperimentazione ma il passaggio dalla didattica dall'applitmica a quello dell'informatica ha un costo troppo elevato per i singoli insegnanti in assenza di materiali di studio in lingua italiana di elevata qualità, auspicabilmente nella forma di Open Educational Resource (OER).

*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]

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
4	Conclusioni	9

1 Introduzione

Gli insegnanti di informatica italiani dei licei italiani, a differenza dei colleghi di molte altre nazioni, non hanno sillabi, programmi nazionali e sistemi di valutazione sviluppati per l'insegnamento della loro materia ma devono progettare e programmare l'attività didattica mediando fra vincoli normativi nazionali, quadri europei sulla competenza digitale⁴ e altre disparate esigenze.

La programmazione didattica disciplinare si realizza come se si dovesse ottimizzare un modello di programmazione matematica mai formalizzato. Le variabili decisionali sono gli argomenti e i contenuti utili a sviluppare le competenze proprie dell'informatica. I vincoli sono dati dalle Indicazioni nazionali [9] e dal Piano Triennale dell'Offerta Formativa della scuola. La funzione obiettivo pondera le necessità del territorio, l'adeguamento alle aspettative di studenti, genitori e altri insegnanti, la presenza dei contenuti del libro di testo in adozione, laddove questo ci sia, il riuso del materiale didattico già realizzato, l'incentivo al conseguimento di certificati, rilasciati a pagamento, sull'uso di software applicativi, specie nel caso in cui la scuola sia sede d'esame, la conformità ai *framework di competenze*, e le conoscenze degli insegnanti stessi⁵.

Nelle esperienze dell'autore, l'aspettativa dell'utenza e del corpo docente si concentra verso un percorso di applicativa* volto a fornire agli studenti le sole abilità d'uso di qualche programma applicativo con il quale interagire tramite un'interfaccia grafica e un comportamento *What You See Is What You Get*⁶ (WYSIWYG). Nel dettaglio, la classe dei programmi applicativi richiesti dagli insegnanti, include sempre un certo programma per l'elaborazione del testo e un altro per la predisposizione dei lucidi di supporto alla presentazione. In aggiunta possono essere aggiunti i fogli elettronici e alcuni sistemi per la rappresentazione geometrica e/o per il calcolo numerico o simbolico.

⁴Si veda il DigComp 2.2 con oltre 250 competenze digitali. L'aggettivo *digitale* probabilmente si riferisce alle dita che esercitano pressione sulla tastiera piuttosto che essere la traduzione del falso amico che in passato si traduceva con *numerico*.

⁵Il 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 classi di concorso e molti insegnanti, specie quelli meno giovani, non hanno sostenuto alcun esame di informatica nel percorso universitario.

⁶Questo 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.

I libri di testo del biennio sono spesso adatti a soddisfare questo genere di aspettative, anche per rispettare le Indicazioni nazionali, e negli ultimi tre anni presentano un linguaggio di programmazione⁷, spesso uno solo e frequentemente il C++⁸.

Nelle nazioni anglofone, le uniche per le quali l'autore è riuscito ad effettuare una ricerca sullo stato dell'arte, esistono vari curricula molto ben dettagliati che fanno apparire il nostro processo di programmazione didattica disciplinare empirica, eterogenea, orientata all'uso delle applicazioni e non supportata da materiale di studio valido⁹.

Ad esempio, negli Stati Uniti ci sono 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 quali [12] e 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 programmazione¹⁰ e ambienti per l'insegnamento¹¹ ([1] e [2], [6]). Anche in Gran Bretagna c'è molta attenzione all'insegnamento dell'informatica e alla formazione professionale dei docenti¹² [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 lingue¹³.

Questo documento presenta nel par. 2 la proposta di curriculum che l'autore ha progettato riducendo il peso delle scelte relative all'applitmatica, nel successivo par. 3 si illustra la sperimentazione e si chiariscono le cause della mancata osservazione comparativa con altre classi del medesimo ciclo e nel par. 4 si determinano le condizioni dell'applicabilità del curriculum in base alla sperimentazione.

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)

⁷Nel primo biennio i linguaggi di programmazione sono sostituiti da ambienti grafici per la programmazione visuale a blocchi e il termine *programmazione* è sostituito da *coding*

⁸Bjarne 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.

⁹Il 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 vera per l'informatica.

¹⁰Albeson 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.

¹¹Seymour 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.

¹²Un progetto di riferimento è [4] ma tante comunità e fondazioni di produttori di hardware Open Source che propongono curricula completi.

¹³La guida è tradotta in tedesco, spagnolo e polacco. Alla lista manca l'italiano e non sono segnalati progetti di traduzione nella lingua italiana.

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
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 eseguito¹⁴ 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.

¹⁴Si 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 con il risolutore
5. Descrivere l'ordine delle operazioni nell'aggiornamento di un foglio di calcolo

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

Nell'incapacità dell'autore di decodificare le richieste di *implementazione di un linguaggio di programmazione, metodologie di programmazione, sintassi di un linguaggio orientato agli oggetti*, si è scelto di presentare il paradigma della programmazione strutturata (con funzioni definite nell'ambito di una funzione) in JavaScript, quello logico con esempi in Prolog, e quello funzionale in JavaScript, realizzando funzioni pure di ordine superiore.

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 all'albero sintattico di una pagina HTML
3. Strutturare i dati per permettere la costruzione programmatica di una pagina HTML
4. Realizzare una semplice web app interattiva

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 un un linguaggio di programmazione e al foglio di calcolo il metodo di Newton per l'ottimizzazione
5. Realizzare un un linguaggio di programmazione e al foglio di calcolo il metodo della sezione aurea

3 Sperimentazione

L'azione didattica ispirata da tale curriculum è stata condotta tra gli anni scolastici 2018/19 e 2022/23 coinvolgendo due classi, di cui una di una sperimentazione quadriennale. Entrambe le classi hanno avuto come insegnante, dal primo all'Esame di Stato, l'autore di questo contributo.

Le misure adottate per fronteggiare l'emergenza sanitaria per COVID-19 nel periodo in cui si è adottato il curriculum hanno condizionato negativamente sulla sperimentazione e sulla rilevazione dei risultati. È stato impossibile condurre un'analisi sul raggiungimento degli obiettivi di competenza per classi parallele al termine del secondo anno, periodo in cui gli studenti non erano nelle aule scolastiche, e non si è proceduto al rilevamento nel quarto e quinto anno per via della forte eterogeneità con le classi parallele che si è avuta anche per l'assenza di didattica.

L'esperienza si è conclusa per l'esigenza di uniformare l'offerta didattica tra classi parallele, per la difficoltà di reperire validi supporti didattici e per la scarsità del tempo che l'autore ha trovato per la produzione o la traduzione del materiale didattico.

4 Conclusioni

In conclusione si considera il curriculum qui proposto come una proposta coerente con i traguardi di competenze fissate dal Ministero dell'Istruzione e del Merito che include alcuni elementi di dettagli della programmazione didattica orientati alla disciplina scientifica piuttosto che all'uso delle applicazioni. Tale curriculum può essere una bozza per la progettazione e la programmazione della didattica dell'informatica alla quale gli insegnanti liceali possono attingere per la costruzione del percorso del proprio istituto.

Il curriculum presenta alcune difficoltà di realizzazione in particolare per la mancanza di materiali di studio, auspicabilmente nella forma di Open Educational Resource (OER), in lingua italiana e di qualità elevata.

Questa difficoltà è superabile se gli insegnanti partecipano a comunità professionali che arricchiscano di OER i progetti esistenti, come quello del Politecnico di Torino [13]. Oltre agli OER servono libri di testo e piattaforme didattiche. Testi e piattaforme ci sono già ma occorre un impegno della comunità nella traduzione professionale dei migliori progetti didattici già attuati all'estero.

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/>.