

UNA "BUONA PRATICA" PER LA FORMAZIONE INFORMATICA NELLE SCUOLE SECONDARIE SUPERIORI

E. Nardelli^{1, 2}

¹ Università di Roma "Tor Vergata"

² CINI – Consorzio Interuniversitario Nazionale per l'Informatica
nardelli@mat.uniroma2.it

Abstract

In questo articolo si presenta il corso "Principi dell'Informatica" che il progetto "Programma il Futuro" propone all'attenzione delle scuole secondarie superiori, per una formazione iniziale all'informatica. Il corso è basato sul materiale del corso *Computer Science Principles* realizzato da Code.org negli Stati Uniti, dove è riconosciuto come valido per il cosiddetto *Advanced Placement*. Argomentiamo che i corsi di laurea in informatica ed ingegneria informatica degli atenei italiani potrebbero trovare vantaggio nell'interagire con le scuole per supportare la realizzazione di questo corso. La sua impostazione, infatti, permette di implementare un percorso basato sull'approccio *flipped classroom* che consentirebbe con uno sforzo ragionevole di effettuare un'attività di formazione efficace e con positivi effetti in materia di orientamento universitario.

Keywords: Scuola secondaria superiore, Formazione informatica, Flipped classroom.

1 INTRODUZIONE

L'informatica è ormai da tempo diventata una disciplina riconosciuta a livello universitario, autonoma e distinta da altre discipline dello stesso rango quali matematica, fisica, chimica, storia e geografia. Alla pari con la matematica ed altre discipline scientifiche, l'informatica è basata su un insieme ben definito di concetti ed approcci ormai stabilizzati, quali, ad esempio, gli algoritmi, le strutture di dati, la programmazione.

Parallelamente con la pervasività delle tecnologie digitali nella società contemporanea, l'informatica sta offrendo nuovi ed importanti modi di osservare e comprendere il mondo in cui viviamo. A tal proposito è spesso usato il termine "computational thinking" [22] (reso in italiano con "pensiero computazionale") per caratterizzare questo nuovo modo di descrivere sistemi naturali (per esempio i sistemi viventi) ed artificiali (per esempio le reti di relazioni sociali).

L'informatica è anche una disciplina che permette di esprimere un alto livello di creatività ed un elevato valore estetico. Ciò è direttamente evidente nel caso dei video-giochi e dell'arte elettronica. In effetti, la creazione di qualunque programma o sistema informatico richiede di esercitare creatività nell'universo virtuale costituito dalle rappresentazioni digitali. Il fatto che questo venga ignorato nella scuola è un elemento su cui intervenire.

Attualmente si insegnano nelle scuole le basi della fisica e biologia, anche se solo una piccola parte di tutti gli studenti diventeranno fisici e biologi, perché viviamo in un mondo basato su questi principi ed è essenziale per ogni studente capirli. Analogamente per l'informatica, è necessario che i nostri studenti capiscano il mondo digitale che li circonda ed agiscano attivamente in esso, senza essere consumatori passivi di una tecnologia per loro opaca.

Una solida comprensione dell'informatica, intesa soprattutto nei suoi aspetti scientifici, permetterà loro di usare con miglior cognizione di causa la tecnologia digitale che correntemente pervade ogni manufatto. Inoltre, cittadini che hanno acquisito le basi scientifiche dell'informatica sono in grado di comprendere e di discutere in modo razionale ed informato fondamentali problemi per la società contemporanea quali quelli del furto d'identità, dell'ingegneria genetica, dei brevetti software, delle votazioni elettroniche, solo per citare alcuni tra i più importanti.

Pertanto, è opportuno che anche nella scuola – come per altre discipline scientifiche che sono state introdotte nel corso del secolo passato – vi sia una formazione di base sull'informatica [21].

Questo è, tra l'altro, in linea con le tendenze di molti paesi occidentali: USA [1], Regno Unito [2], Francia [3], sono alcuni tra i più importanti che si sono già avviati su questa strada.

Nell'ambito del progetto "Programma il Futuro" [4] già da due anni stiamo proponendo a tutte le scuole italiane percorsi introduttivi di informatica basati su personaggi di film e giochi di elevata popolarità tra i

ragazzi e focalizzati sui concetti chiave dell'informatica più che sui dettagli tecnologici o dello specifico linguaggio di programmazione [5]. Il progetto è realizzato dal CINI (Consorzio Interuniversitario Nazionale per l'Informatica) in base ad un accordo col MIUR (Ministero dell'Istruzione, Università e Ricerca)

Un'eccellente opportunità per le scuole secondarie superiori si rende adesso disponibile con il corso "Principi dell'Informatica". Il corso è basato sul materiale del corso "Computer Science Principles" realizzato da Code.org negli Stati Uniti, dove è riconosciuto come valido per il cosiddetto "Advanced Placement". Tale meccanismo permette agli studenti degli ultimi anni dei licei di acquisire crediti formativi validi a livello universitario in vari settori, da quelli artistico e linguistici a quelli delle varie discipline scientifiche [6].

Nel seguito presentiamo organizzazione e contenuti di tale corso, che è completamente accessibile via rete, argomentando che si tratta di una vantaggiosa possibilità per i corsi di laurea in informatica ed ingegneria informatica degli atenei italiani di portare una formazione informatica di elevata qualità scientifica nelle scuole secondarie superiori, con potenziali effetti assai benefici in termini di orientamento e di diffusione di cosa sia realmente l'informatica.

2 OBIETTIVI E PREREQUISITI

Il corso sviluppa un percorso narrativo basato su Internet e l'innovazione, come temi che connettono tutte e sei le unità del corso stesso. Si inizia da come inviare un bit da un posto ad un altro e si finisce col riflettere sulle implicazioni di un'innovazione digitale. Lungo questo percorso si impara:

- come funziona Internet ed il suo impatto sulla società;
- come programmare e prototipare rapidamente semplici applicazioni per il web (in JavaScript) che risolvono problemi reali;
- come raccogliere, analizzare e visualizzare dati per ottenere conoscenza e comprensione su fenomeni sociali e naturali;
- come valutare gli effetti sia positivi che negativi delle innovazioni digitali su persone e società.

Per un adeguato svolgimento del corso è richiesta agli studenti la conoscenza dell'algebra che si acquisisce normalmente con i primi due anni del liceo e la maturità linguistico-espressiva conseguita al termine dello stesso periodo.

Inoltre, non è necessaria alcuna conoscenza pregressa dei concetti dell'informatica ed il corso è fruibile anche da studenti (e docenti) che sono nuovi della disciplina. Attività didattiche, video di supporto, strumenti informatici disponibili nel corso sono attraenti ed accessibili a studenti e docenti delle più varie provenienze, esperienze, ed interessi. Tutte queste risorse sono state realizzate in modo da permettere al docente di concentrarsi sul ruolo di facilitatore e tutore per lo studente piuttosto che sul presentare ed illustrare direttamente i vari concetti.

Pertanto il corso può essere insegnato anche da docenti che non hanno una specifica formazione universitaria sull'informatica, purché si trovino a proprio agio con la tecnologia digitale (PC e tablet, browser e posta elettronica, download di file) ed abbiano un minimo di esperienza di programmazione. Il progetto "Programma il Futuro" offre a questo specifico scopo il "Corso Rapido" [8] che realizza una veloce introduzione ai concetti fondamentali del pensiero computazionale: ripetizione, selezione, variabili, funzioni, senza dimenticare l'importanza della verifica e della correzione degli errori. Il tutto mediante un linguaggio visuale a blocchi ed in un contesto estremamente amichevole per l'utente.

Da un punto di vista di infrastrutture tecnologiche, il corso richiede che ogni studente abbia la disponibilità di un PC connesso ad Internet e con un browser moderno installato. Tutte le risorse didattiche utilizzate durante il corso sono disponibili online ed utilizzabili mediante il browser.

A seconda dell'organizzazione didattica e logistica, tale PC può essere quello del laboratorio della scuola o quello di casa.

3 IMPOSTAZIONE, ORGANIZZAZIONE E STRUTTURA DELLE LEZIONI

Le lezioni sono state realizzate in modo da essere "centrate sullo studente", che viene quindi stimolato a procedere con attività basate su indagini finalizzate a scoprire e comprendere i concetti. Non è quindi necessario che l'insegnante svolga un'attività didattica tradizionale dalla cattedra. Le istruzioni dirette per l'uso del materiale didattico sono contenute all'interno del materiale stesso e lo studente può con profitto iniziare a lavorare su ogni lezione a casa, mentre il tempo in aula viene destinato alla discussione e al confronto.

Questa impostazione di "insegnamento capovolto" (*flipped classroom*) rende il corso particolarmente adatto per essere inserito anche all'interno di un'organizzazione didattica normalmente rigida quale quella esistente con gli attuali quadri orari della scuola secondaria superiore. È inoltre possibile dispiegare lo svolgimento di un tale corso anche durante più anni scolastici, in funzione del numero di ore settimanali che si riesce ad assegnargli.

Le 6 unità sono:

1. Internet (14 lezioni, divise in due capitoli)
2. I dati digitali (15 lezioni, divise in due capitoli)
3. Algoritmi e Programmazione (10 lezioni)
4. I *big data* e la *privacy* (9 lezioni)
5. La realizzazione di *app* (21 lezioni)
6. Prova finale (4 lezioni)

Ognuna delle prime cinque unità "sviluppa una storia" relativa ad uno specifico tema dell'informatica, partendo da un primo accenno fino ad un più articolato sviluppo conclusivo. In particolare, le prime tre unità sono dedicate ai Concetti Fondamentali che sono specifico oggetto di studio del corso (Internet, i Dati, Algoritmi e Programmazione). La quarta unità approfondisce il tema dei *big data* e del loro impatto globale sulla società, mentre la quinta unità approfondisce il tema della programmazione per lo sviluppo di applicazioni. La sesta unità è dedicata ad attività progettuale di preparazione all'esame finale.

Ogni unità è organizzata in un numero variabile di lezioni, ognuna delle quali presuppone la conoscenza dei concetti e l'acquisizione delle abilità oggetto delle lezioni precedenti. Ogni lezione ha l'obiettivo di condurre lo studente da una premessa o da un'esigenza verso lo svolgimento di un'attività che gli fa acquisire conoscenza ed abilità relative ad uno (o più) specifico obiettivo di apprendimento.

La maggior parte delle lezioni è così strutturata:

- un'attività introduttiva
- un'attività sostanziale
- un'attività o riflessione di sintesi finale

Ogni unità contiene almeno una valutazione sommativa o un progetto, che hanno lo scopo di preparare lo studente allo svolgimento della prova finale.

L'organizzazione del corso è descritta più in dettaglio in [8]. Le unità complessivamente comprendono circa 75 lezioni da circa 2 ore l'una. È da aggiungere a questo il tempo necessario lo svolgimento degli esercizi e progetti (che ovviamente dipende dalle capacità del singolo studente) e per le discussioni ed i confronti in aula. Stimiamo che, includendo questo tempo, ogni lezione richieda allo studente un impegno complessivo tra le 3 e le 4 ore, per un totale di 200-300 ore.

Nella pianificazione descritta nel materiale originale in inglese è previsto di svolgere il corso lungo un arco di 30 settimane di uno stesso anno scolastico. Questo è possibile in virtù dell'organizzazione didattica presente negli Stati Uniti. In Italia il quadro orario settimanale della scuola secondaria superiore è molto più rigido e un'attività di questo genere deve necessariamente esser fatta rientrare o in progetti didattici specifici della scuola oppure – data la recente introduzione della norma relativa all'Alternanza Scuola-Lavoro [9, 10] – essere svolta nell'ambito di un progetto di tal tipo. Ricordiamo che tale norma prevede che tutti gli studenti della secondaria superiore svolgano negli ultimi tre anni un monte ore complessivo in attività di questo tipo, stabilito in almeno 400 ore per gli istituti tecnici e professionali e in almeno 200 ore per i licei.

In base alla normativa le università sono soggetti titolati di diritto ad erogare per gli studenti delle scuole percorsi di alternanza scuola lavoro. Si tratta di un'eccellente opportunità per sopperire da subito alla carenza degli attuali programmi scolastici che non riconoscono ancora l'informatica come materia importante nell'istruzione obbligatoria. In aggiunta, i Consigli di Corso di Laurea in Informatica e Ingegneria Informatica potrebbero decidere di riconoscere agli studenti che hanno svolto tale percorso dei crediti validi ai fini della carriera universitaria, esattamente come accade nel contesto dell'*Advanced Placement* negli Stati Uniti.

Ovviamente rimane sempre ferma la possibilità di svolgere tale percorso nell'ambito di un accordo di alternanza con un'azienda interessata al tema dello sviluppo di competenze digitali.

4 QUADRO DI RIFERIMENTO DIDATTICO

Il corso è stato realizzato attorno a sette "**Concetti Fondamentali**" dell'Informatica (*Big Ideas*) ed a sei "**Competenze del Pensiero Computazionale**" (*Computational Thinking Practices*) e garantisce che gli studenti acquisiscano al tempo stesso sia i concetti che le competenze.

I sette Concetti Fondamentali sono:

1. La *Creatività* e l'informatica sono importanti forze propulsive dell'innovazione. Le innovazioni rese possibili dall'informatica hanno avuto e continueranno ad avere impatti estesi e di lunga durata.
2. L'*Astrazione* riduce la quantità di dati e di dettagli da trattare, facilitando la concentrazione sugli aspetti più rilevanti. Si tratta di un processo, di una strategia, che permettono di meglio comprendere e risolvere un problema.
3. I *Dati* sono essenziali per la creazione di conoscenza. L'informatica realizza metodi efficienti per la loro elaborazione, rendendo così possibili cambiamenti impressionanti in tutte le discipline, dall'arte all'economia alla scienza.
4. Gli *Algoritmi* vengono usati per sviluppare efficienti soluzioni operative a problemi risolvibili mediante elaborazione di dati. Espresi nei programmi informatici hanno cambiato il mondo in modo profondo e durevole.
5. La *Programmazione* rende possibile la risoluzione dei problemi, la creazione di conoscenza e l'espressione umana. I programmi danno luogo a sistemi e strumenti informatici che facilitano la creazione di artefatti digitali, quali musica, immagini, visualizzazioni.
6. *Internet* pervade il moderno panorama digitale: la sua diffusione ha cambiato radicalmente la società negli ultimi vent'anni, rendendo possibili modalità di comunicazione, interazione e collaborazione del tutto nuove.
7. L'*Impatto Globale* dell'informatica è sotto gli occhi di tutti. Le interazioni sociali, il mondo degli affari e della produzione, la risoluzione dei problemi, sono stati cambiati dalle innovazioni dell'informatica, e cambieranno ancora.

Tali concetti non sono disgiunti ma presentano intersezioni e riferimenti incrociati.

Le sei Competenze del Pensiero Computazionale sono:

1. La capacità di *Connettere* e di mettere in relazione i differenti concetti dell'informatica.
2. La competenza nel *Creare Artefatti Computazionali* permette agli studenti di sviluppare creatività, lavorando alla progettazione e sviluppo di artefatti digitali e risolvendo problemi mediante l'utilizzo di tecniche informatiche.
3. Gli studenti imparano ad *Astrarre* per definire modelli e simulazioni di fenomeni naturali e artificiali, fare predizioni sulla loro evoluzione ed analizzarne efficacia e validità.
4. L'abilità di *Analizzare Problemi ed Artefatti*, sia realizzati in prima persona che sviluppati dagli altri, è essenziale affinché gli studenti progrediscano nella capacità di risoluzione dei problemi.
5. Il saper *Comunicare* è essenziale per lo studente, sia per spiegare e giustificare le scelte progettuali e realizzative degli strumenti informatici che produce, sia per analizzare e valutare i risultati ottenuti alla scelta delle esigenze iniziali.
6. Gli studenti imparano a *Collaborare* in molte attività, sia quelle investigative relative ai dati ed alle loro relazioni che quelle progettuali relative alla realizzazione degli artefatti digitali.

Tali competenze sono abilità di alto livello che vanno sviluppate ed approfondite nel tempo.

I sette Concetti Fondamentali rappresentano un corpo di conoscenze in cui quattro specifici oggetti di studio (Internet, i Dati, gli Algoritmi, la Programmazione) si intersecano con tre principi più generali dell'informatica (Creatività, Astrazione, Impatto Globale).

Ognuno degli specifici obiettivi di apprendimento, che sono discussi in modo dettagliato e completo nella guida [11] insieme alla descrizione delle modalità di esame previste negli Stati Uniti, fa normalmente riferimento ad uno o più dei Concetti Fondamentali e contemporaneamente mette in gioco una o più delle Competenze del Pensiero Computazionale. Per ognuno dei Concetti Fondamentali sono elencati uno o più "**elementi durevoli di comprensione**" (*Enduring Understandings*: lo studente comprende che...) ai quali corrispondono uno o più "**obiettivi di apprendimento**" (*Learning Objectives*: lo studente è in grado di...) e, corrispondentemente, uno o più "**elementi essenziali di conoscenza**" (*Essential Knowledge*: lo studente sa che...).

5 CONTENUTI DIDATTICI

Gli insegnanti hanno a disposizione, per ogni unità, il piano didattico che descrive in dettaglio, lezione per lezione, i concetti da studiare e le attività da svolgere. Vi sono inoltre esempi di valutazione (sia formativa che sommativa), guide per lo svolgimento delle varie attività, strumenti di controllo dei progressi di ogni studente, video esplicativi.

Analogamente, per ogni unità gli studenti hanno a disposizione il materiale didattico necessario per lo studio, in completa autonomia, degli argomenti oggetto di ogni lezione: testi e materiale esplicativo, guide allo svolgimento delle attività, video illustrativi, strumenti informatici interattivi di sperimentazione e di programmazione.

Tutto il materiale è accessibile via Internet, e il docente può gestire la sua classe attraverso un cruscotto di controllo che gli permette di monitorare l'avanzamento dello studio di ogni studente, somministrare prove di valutazione e ricevere le risposte.

In particolare, gli strumenti informatici disponibili [12, 13] sono:

- un compressore di testi,
- due gestori di pixel, uno per pixel in bianco/nero ed uno per il colore
- due strumenti di crittografia
- un simulatore di Internet
- l'ambiente AppLab per la realizzazione di App in JavaScript

Come materiale di lettura supplementare viene anche indicato come disponibile per il download gratuito il libro *Blown to Bits* [14].

Esempi delle prove finali sono descritti nella pubblicazione sopra citata che descrive il quadro di riferimento didattico.

5.1 Unità 1: Internet

Questa unità [15] esplora i problemi e le sfide tecniche che derivano dalla necessità di rappresentare i dati in formato digitale per consentire al computer la loro elaborazione e trasmissione.

Nel primo capitolo (Rappresentare e trasmettere l'informazione) si analizza cosa è necessario per inviare un dato elementare (bit) da un posto ad un altro, e la complessità aumenta gradualmente, man mano che gli studenti lavorano sulle problematiche dell'invio di dati con più bit e maggiore struttura. Gli studenti usano il Simulatore di Internet, che gli permette di sperimentare con le codifiche binarie dei dati e con i protocolli di comunicazione usati per la loro trasmissione.

Gli studenti proseguono nel secondo capitolo (Inventare Internet) affrontando lo stesso tipo di problemi che sono stati risolti nella costruzione di Internet e sviluppando i loro protocolli di comunicazione, ciascuno basato sul precedente, ed arrivando ad una loro stratificazione, come accade nella rete reale.

Il progetto svolto nell'unità richiede allo studente di investigare un problema sociale attuale legato ad Internet quale, ad esempio, la neutralità della rete o la censura sulla rete.

L'indice delle lezioni è il seguente:

Capitolo 1: Rappresentare e trasmettere l'informazione

1. Innovazioni personali
2. Invio di messaggi binari
3. Invio di messaggi binari col Simulatore di Internet
4. Sistemi di numerazione
5. La numerazione binaria
6. Invio di numeri
7. Codifica ed invio di testo

Capitolo 2: Inventare Internet

8. Internet
9. La necessità di gestire indirizzi
10. Instradamento dei messaggi e ridondanza
11. I pacchetti e l'affidabilità di Internet
12. La necessità di un sistema per la gestione dei nomi (DNS)
13. Astrazione e protocollo HTTP
14. Progetto: Internet e la società

L'unità offre anche cinque lezioni opzionali di approfondimento: due sono dedicate ad esaminare come nel mondo reale si inviano i bit e si codificano i numeri, mentre tre esplorano algoritmi necessari per il funzionamento di Internet (il cammino minimo, il minimo albero di connessione, l'instradamento dei messaggi).

5.2 Unità 2: I dati digitali

Questa unità [16] approfondisce il tema della rappresentazione ed elaborazione digitale dei dati. Gli studenti imparano a generare, ripulire, ed elaborare dati, utilizzando poi strumenti di visualizzazione per esplorare ed identificare tendenze e schemi ricorrenti.

Il capitolo 1 dell'unità (Codificare e comprimere dati complessi) riparte dalla rappresentazione binaria dei dati ed analizza varie tecniche utili per risparmiare lo spazio necessario per la rappresentazione di testo ed immagini (tecniche di compressione). Un primo progetto richiede agli studenti di definire il loro schema di compressione dei dati per una loro esperienza significativamente complessa.

Successivamente, gli studenti si esercitano nel Capitolo 2 (Usare e visualizzare i dati) nell'interpretazione di dati visuali e nell'utilizzo di strumenti per creare i loro personali artefatti digitali. In questo processo, imparano a raccogliere e ripulire i dati, e ad utilizzare strumenti per aggregarli e visualizzarli. Riflettono inoltre sul ruolo dell'errore umano durante la raccolta e l'analisi e di come questo possa portare a conclusioni inaccurate e potenzialmente dannose.

Nel progetto finale gli studenti devono lavorare in collaborazione su un insieme di dati grezzi, usando strumenti per individuare possibili relazioni e tendenze, realizzare un spiegazione visuale di quanto hanno trovato, e scrivere un documento illustrativo.

L'indice delle lezioni è il seguente:

Capitolo 1: Codificare e comprimere dati complessi

1. I byte e le dimensioni dei file
2. Compressione di testi
3. Codifica di immagini in bianco e nero
4. Codifica di immagini a colori
5. Compressione con perdite e formati di file
6. Progetto: codifica di un'esperienza

Capitolo 2: Usare e visualizzare i dati

7. Introduzione ai dati
8. Scoperta di tendenze attraverso la visualizzazione
9. Verifica delle proprie ipotesi
10. Visualizzazioni buone e cattive

11. Realizzazione di visualizzazioni dei dati
12. Scoperta di una storia nei dati
13. Pulizia dei dati
14. Creazione di tabelle di sintesi
15. Progetto: il racconto di una storia relativa ai dati

5.3 Unità 3: Algoritmi e programmazione

Questa unità [17] introduce la necessità degli algoritmi ed avvia lo studente al linguaggio di programmazione JavaScript, insegnandogli a sviluppare le prime semplici applicazioni fruibili attraverso il Web (le *app*).

Si inizia con attività tradizionali (senza utilizzare i PC) per introdurre gli algoritmi e mostrare la necessità di un linguaggio di programmazione che renda possibile eseguirli su un PC. In questo percorso si stabiliscono connessioni con le regole definite nei protocolli per Internet discussi nell'unità 1, nei quali erano stati gli studenti ad agire come computer. Questo permette di comprendere come l'essenza di un programma informatico sia una serie di passi che una macchina digitale è in grado di compiere autonomamente e meccanicamente.

Gli studenti vengono introdotti all'uso dell'ambiente di programmazione AppLab, in cui scrivono programmi che controllano una "tartaruga", personaggio virtuale che si muove sullo schermo e realizza disegni. Lo studio di JavaScript è centrato sullo sviluppo di procedure (o funzioni, cioè porzioni di programmi riutilizzabili) in modo da abituare lo studente ad un approccio top-down nella realizzazione dei programmi.

Col progetto finale gli studenti collaborano per progettare e condividere codice che crea, mediante la tartaruga, una storiella disegnata.

L'indice delle lezioni è il seguente

1. La necessità dei linguaggi di programmazione
2. La necessità degli algoritmi
3. Creatività negli algoritmi
4. L'uso di semplici comandi
5. La creazione di funzioni
6. Funzioni e progettazione top-down
7. Le interfacce di programmazione (API) e l'uso di funzioni con parametri
8. La creazione di funzioni con parametri
9. I cicli ed i numeri casuali
10. Progetto: realizzare una storia digitale

5.4 Unità 4: I *big data* e la *privacy*

In questa unità [18] gli studenti sviluppano una visione complessiva e bilanciata sui dati nel mondo intorno a loro e sui loro effetti positivi e negativi. Inoltre, studiano le basi del funzionamento delle moderne tecniche crittografiche.

L'unità inizia facendo riflettere gli studenti sull'enorme quantità di dati digitali (*big data*) del mondo moderno, sul ruolo essenziale che hanno i computer nella loro elaborazione, e sui possibili usi sia positivi (ad esempio, per sostenere l'innovazione) che negativi (ad esempio, per controllare e censurare le persone).

Si prosegue con lo studio della crittografia, partendo da un'attività di tipo tradizionale (senza computer) e proseguendo con gli strumenti crittografici forniti per esplorare ulteriormente il concetto. In tal modo gli studenti riescono a comprendere le varie idee matematiche alla base dello sviluppo delle tecniche di crittografia e come/quando tali tecniche possono essere violate. In questo processo, acquisiscono consapevolezza dell'esistenza di problemi "computazionalmente difficili". Alla fine, lo studente acquisisce una conoscenza di alto livello della crittografia a chiave asimmetrica, di cosa essa rende

possibile (inviare dati crittati senza condividere una chiave) e di cosa rende sostanzialmente impossibile (violare i dati crittati).

Il progetto finale è focalizzato su di un argomento di attualità relativo ai *big data*, la sicurezza e la crittografia.

L'indice delle lezioni è il seguente:

1. Cosa sono i *big data*?
2. Ricerca: le innovazioni basate sui dati
3. Identificazione delle persone mediante i dati
4. Il costo reale di quello che è gratis
5. Crittare in modo semplice
6. Crittare con chiavi e password
7. La crittografia a chiave pubblica
8. Ricerca: il crimine informatico (*cybercrime*)
9. Progetto: i *big data* e i dilemmi della sicurezza

L'unità offre anche due lezioni opzionali di approfondimento: una che prende in esame quello che è forse il più famoso dei problemi "computazionalmente difficili" (il commesso viaggiatore) ed un'altra che esplora le relazioni tra questo tipo di problemi e le funzioni unidirezionali (cioè le funzioni facili da calcolare ma difficili da invertire).

5.5 Unità 5: La realizzazione di App

Questa unità¹ [19] approfondisce lo studio del linguaggio di programmazione JavaScript e lo studente impara a realizzare *app* guidate dagli eventi. L'unità presuppone l'acquisizione di concetti e competenze trattati nell'unità 3: scrivere ed usare le funzioni, l'utilizzo dei cicli, leggere la documentazione, usare l'ambiente AppLab, la collaborazione.

Si comincia usando in AppLab la "modalità di progettazione" che permette agli studenti di prototipare rapidamente un'*app* basata su eventi. Man mano che gli studenti realizzano applicazioni che reagiscono alle azioni degli utenti, vengono esaminati ed acquisiti concetti chiave della programmazione: le variabili, la logica booleana, le istruzioni condizionali.

Si prosegue approfondendo ulteriori concetti chiave, quali i cicli, i vettori, le simulazioni. Gli studenti imparano a valutare la correttezza di un programma e le modalità di realizzazione che rendono più facile tale compito e la futura evoluzione del programma stesso.

L'unità comprende due progetti. Il primo (Assistente Digitale) è centrato sulla realizzazione di un video di presentazione di un programma sviluppato dallo studente e sulla descrizione (orale o scritta) di un algoritmo. Nel secondo (Migliora un'*app*) lo studente riprende una delle *app* sviluppate nel corso dell'unità ed identifica, progetta e realizza i suoi possibili miglioramenti.

L'indice delle lezioni è il seguente:

1. Gli eventi
2. Programmazione guidata dagli eventi e correzione degli errori
3. Oltre i pulsanti verso le *app*
4. Introduzione alla "modalità di progettazione"
5. *App* multi-schermo
6. Il controllo della memoria mediante le variabili
7. L'uso delle variabili nei programmi
8. L'input dell'utente e le stringhe
9. Introduzione al progetto "Assistente Digitale"
10. Capire il flusso e la logica di un programma
11. Introduzione alle condizioni logiche
12. Condizioni logiche complesse
13. Progetto: Assistente Digitale
14. I cicli "mentre"

¹ Questa unità è in revisione: la versione aggiornata sarà disponibile all'inizio di Dicembre 2016.

15. Cicli e simulazioni
16. Introduzione ai vettori
17. Scorrimento di immagini mediante eventi
18. Elaborazione di vettori
19. Funzioni che ritornano valori
20. Le "superfici" nelle *app*
21. Progetto: Migliora un'*app*

5.6 Unità 6: La prova finale

L'unità² [20] è dedicata alla realizzazione da parte dello studente dei due progetti finali. Un progetto è dedicato alla "Creazione" di un programma informatico, mentre l'altro è dedicato alla "Esplorazione" di un'innovazione dell'informatica. Nel caso il corso venga sviluppato nell'ambito di un accordo con un ateneo (che sia nel contesto dell'Alternanza Scuola-Lavoro o meno), il superamento di questa prova servirà per acquisire i crediti universitari eventualmente riconosciuti dal relativo Consiglio di Corso di Laurea.

Il coinvolgimento del docente è minimale e focalizzato essenzialmente ad assicurare che lo studente abbia obiettivi ben chiari e piani realizzativi accettabili.

Il progetto di "Creazione" deve avere dimensioni tali da poter essere realizzato in autonomia dallo studente in circa 12 ore, mentre quello di "Esplorazione" dovrà richiedere allo studente di lavorare in autonomia per circa 8 ore.

L'indice delle lezioni è il seguente:

1. Preparazione del progetto "Creazione"
2. Esecuzione del progetto "Creazione"
3. Preparazione del progetto "Esplorazione"
4. Esecuzione del progetto "Esplorazione"

6 CONCLUSIONI

Abbiamo presentato l'impostazione generale, la struttura, il quadro di riferimento didattico e gli specifici contenuti del corso "Principi dell'Informatica", che il progetto "Programma il Futuro" suggerisce per le scuole secondarie superiori come formazione iniziale all'informatica di elevata qualità scientifica e culturale.

Il corso può essere utilizzato dalle scuole sia in piena autonomia, sia con il supporto di aziende (nell'ambito di un accordo di Alternanza Scuola-Lavoro) che in collaborazione con le Università. Anche in quest'ultimo caso il corso potrebbe essere inserito in accordi di alternanza e, in modo indipendente, dare eventualmente luogo (come accade negli Stati Uniti con il meccanismo dell'*Advanced Placement*) al riconoscimento agli studenti di crediti formativi universitari.

In ogni caso, riteniamo che il corso offra ai Corsi di Laurea in Informatica ed in Ingegneria Informatica un'eccellente possibilità – con un basso sforzo organizzativo – di svolgere nelle scuole secondarie superiori un'efficace attività di orientamento verso l'informatica e una valida opera di diffusione della cultura informatica.

² Questa unità è in revisione: la versione aggiornata sarà disponibile all'inizio di Dicembre 2016.

REFERENCES

- [1] National Science Foundation, *Initiative "Computer Science For All"*, <http://www.nsf.gov/csforall>, January 2016.
- [2] The Royal Society, *Shut down or restart? The way forward for computing in UK schools*, January 2012.
- [3] Académie des Sciences, *L'enseignement de l'informatique en France: Il est urgent de ne plus attendre*, Mai 2013.
- [4] Progetto "Programma il Futuro", <http://programmairfuturo.it>
- [5] E.Nardelli, G.Ventre, *Introducing computational thinking in Italian schools: a first report on "Programma il Futuro" project*, 9th *International Technology, Education and Development Conference (INTED-2015)*, Madrid, Spain, March 2015, <https://library.iated.org/view/NARDELLI2015INT>
- [6] <https://advancesinap.collegeboard.org/>
- [7] Il "Corso Rapido" di introduzione all'informatica, <http://programmairfuturo.it/come/lezioni-tecnologiche/corso-rapido>
- [8] Code.org, *Syllabus for the AP Computer Science Principles course*, <https://code.org/files/CSPSyllabusApril2016.pdf>
- [9] Ministero dell'Istruzione, dell'Università e della Ricerca, DG per gli ordinamenti scolastici e la valutazione del Sistema nazionale di istruzione, *Attività di alternanza scuola-lavoro: guida operativa per la scuola*, <http://www.istruzione.it/allegati/2015/guidaASLinterattiva.pdf>
- [10] Ministero dell'Istruzione, dell'Università e della Ricerca, *Sito web dell'alternanza scuola-lavoro*, <http://www.istruzione.it/alternanza/index.shtml>
- [11] College Board, *AP Computer Science Principles 2016-17: course and exam description, including the curriculum framework*, <https://secure-media.collegeboard.org/digitalServices/pdf/ap/ap-computer-science-principles-course-and-exam-description.pdf>
- [12] Code.org, *Widgets for the Computer Science Principles course*, <https://code.org/educate/csp/widgets>
- [13] Code.org, *AppLab: a programming environment for the Computer Science Principles course*, <https://code.org/educate/applab>
- [14] H.Abelson, K. Leden, H.Lewis, *Blown to Bits*, Addison Wesley, 2008, <http://www.bitsbook.com/excerpts/>
- [15] Code.org, *Unità 1: Internet*, <https://curriculum.code.org/csp/unit1/>
- [16] Code.org, *Unità 2: I dati digitali*, <https://curriculum.code.org/csp/unit2/>
- [17] Code.org, *Unità 3: Algoritmi e Programmazione*, <https://curriculum.code.org/csp/unit3/>
- [18] Code.org, *Unità 4: I big data e la privacy*, <https://curriculum.code.org/csp/unit4/>
- [19] Code.org, *Unità 5: La realizzazione di app*, <https://curriculum.code.org/csp/unit5v1/>
- [20] Code.org, *Unità 6: La prova finale*, <https://curriculum.code.org/csp/unit6v1/>
- [21] Informatics Europe and ACM Europe: "Informatics education: Europe cannot afford to miss the boat", April 2013, <http://www.informatics-europe.org/images/documents/informatics-education-europe-report.pdf>
- [22] Wing, J. (2006). Computational thinking. *Comm. ACM* 49(3), pp.33-35. <http://www.cs.cmu.edu/~15110-s13/Wing06-ct.pdf>