

Editoriale

DIDAMATICA 2022

La trasformazione digitale nell'ambito "education"

L'accelerazione data dalla pandemia alla trasformazione digitale in tutti i settori è evidente. Il settore pubblico non fa eccezione, inclusa la pubblica amministrazione che ha accelerato un complesso percorso di modernizzazione, peraltro iniziato ben prima della pandemia. Nel caso del mondo education le azioni di trasformazione digitale imposte dalla pandemia hanno prima di tutto permesso forme di continuità didattica impensabili fino a pochi anni fa. Nel giro di pochi giorni scuole e università si sono viste costrette ad usare una varietà di strumenti informatici coinvolgendo studenti e docenti in un gigantesco esperimento di trasformazione digitale. L'uso didattico di strumenti collaborativi nati per altre attività, quali ad esempio Microsoft Teams e Google Meet - ha presentato indubbi vantaggi ma ha mostrato anche parecchi limiti.

In linea generale, l'emergenza pandemica ha arricchito di esperienze su larga scala la complessa relazione tra innovazione tecnologica e mondo della scuola. Accanto al tema dell'efficacia del modello di educazione a distanza, che ha monopolizzato l'attenzione delle famiglie e degli operatori scolastici, si sono innescate accese discussioni su come la tecnologia possa ridefinire le fondamenta stesse del sistema educativo in termini di organizzazione delle forme didattiche e delle metodologie di apprendimento e insegnamento. La trasformazione digitale si regge su quattro pilastri: la raccolta di dati sempre più pervasivi, l'uso di tecnologie hardware e software sempre più potenti, l'atteggiamento positivo e propositivo di tutte le persone coinvolte nelle nuove realtà digitali (che nel mondo education vuol dire studenti, docenti e personale ausiliario), l'avvento di processi inclusivi e sempre più articolati al servizio delle varie comunità del mondo education.

La conferenza Didamatica 2022 si è svolta a Milano nei giorni 10 e 11 novembre 2022. Il tema della conferenza era "la trasformazione digitale nella scuola, negli ITS, nell'Università e nella formazione professionale". La partecipazione di ricercatori e di docenti impegnati attivamente nelle diverse organizzazioni educative ha permesso di costruire una panoramica sulle attività correnti, in

0

1

0

1

0



particolare nell'ambito delle scuole e delle università. Sono stati sottomessi 81 articoli, di cui accettati e presentati circa 70. Di questi ultimi ne sono stati scelti 6 come "best papers", che vengono premiati con la pubblicazione in questo numero speciale di Mondo Digitale. Ringraziamo tutti i membri dei Comitati di Programma, Scientifico e Organizzativo. Ringraziamo altresì i revisori anonimi per il loro paziente e attento lavoro di revisione.

Gli articoli selezionati sono i seguenti:

- **PythagorHub “Quando Pitagora incontra Android”** di Paola Palestini e Luciano Zazzetti. L'articolo presenta un'esperienza di sviluppo di una applicazione di tipo matematico, creata per rendere le lezioni a distanza più coinvolgenti.
- **LaTeX tra competenze digitali e accessibilità: un'esperienza di PCTO con il Laboratorio Polin.** di Tiziana Armano, Carola Manolino, Margherita Piroi, Massimo Borsero, Davide Maietta e Anna Capietto. L'articolo descrive come introdurre gli studenti all'impiego di un linguaggio di marcatura per produrre testi complessi, invitandoli a riflettere sulla struttura astratta del testo.
- **Futuri insegnanti di sostegno. Gli Smart Visual Media per l'inclusione scolastica degli alunni con disabilità** di Pierluigi Muoio. L'articolo presenta una riflessione sulle potenzialità didattiche delle tecnologie di realtà virtuale di tipo immersivo, in particolare per il sostegno didattico di allievi svantaggiati.
- **Sviluppo del pensiero computazionale nella scuola dell'infanzia e nella scuola primaria** di Alice Barana, Francesca Casasso, Cecilia Fissore, Valeria Fradiante e Marina Marchisio. Resoconto di un corso di formazione per insegnanti della scuola dell'infanzia e primaria della Valle d'Aosta sui temi del problem solving e del computational thinking.
- **Di cosa parliamo quando parliamo di 'programmi'** di Violetta Lonati, Claudio Mirolo e Mattia Monga. Questo articolo discute la pratica della programmazione analizzandola per il suo potenziale di strumento di consapevolezza e cittadinanza attiva.
- **Tirocini informatici online in ISTAT: esperienze di learning-by-doing di coding e di utilizzo di strumenti low-code** di Daniela Casale, Maria Pia Di Maio e Giulia Vaste. Questo articolo presenta l'organizzazione di tirocini di studenti di Data Science in ISTAT.

Buona lettura!

Paolo Ciancarini
Alma Mater Studiorum Università di Bologna
Pierfranco Ravotto
AICA

Di cosa parliamo quando parliamo di ‘programmi’*

Violetta Lonati^{1,3}, Claudio Mirolo^{2,3} e Mattia Monga^{1,3}

¹Università degli Studi di Milano

²Università di Udine

³Laboratorio CINI Informatica e Scuola, Italia

Sommario

Il mondo della scuola si sta ormai convincendo che la programmazione debba avere un ruolo sempre più rilevante tra le competenze da acquisire a tutti i livelli e in tutti i percorsi formativi. Del resto è assai opportuno che una parte sempre più ampia della cittadinanza sia in grado di capire cosa significa progettare e realizzare elaborazioni automatizzate. Il rischio, tuttavia, è che la complessità tecnologica spinga a banalizzare gli obiettivi formativi o a soffermarsi su aspetti di dettaglio, perdendo di vista la ricchezza concettuale che la programmazione può dispiegare una volta colte le sue molteplici sfaccettature. Una chiara esposizione degli aspetti chiave dei programmi può aiutare insegnanti e altri operatori culturali a identificare le ragioni della centralità del *software* nella società attuale e a orientare al meglio l'azione educativa, affinché la pratica della programmazione dischiuda tutte le sue potenzialità come strumento di consapevolezza e cittadinanza attiva.

1 La programmazione nella scuola

La programmazione degli “elaboratori di informazioni” sta al cuore dell'informatica. In un certo senso è proprio la ragion d'essere della disciplina, il suo nucleo caratterizzante e motivante l'enorme varietà di approcci e sviluppi che ne vivacizzano lo studio e la pratica. Non stupisce quindi che in una società

* Le riflessioni riportate in questo articolo sono frutto del confronto e della discussione maturata all'interno del gruppo di lavoro WG5, nell'ambito dell'International Conference on Innovation and Technology in Computer Science Education (ITiCSE 2022) di cui fanno parte, oltre ai presenti autori, Tim Bell (NZ), Andrej Brodnik (SLO), Andrew Paul Csizmadia (UK), Liesbeth De Mol (FR), Henry Hickman (NZ), Therese Keane (AU).

dominata dalle tecnologie dell'informazione, la programmazione si stia diffondendo sempre di più anche nelle attività scolastiche, a tutti i livelli e in tutti i percorsi formativi [12, 3]. In realtà c'è forse da chiedersi come mai solo ora si inizi a interrogarsi seriamente su come introdurre compiutamente l'informatica e la programmazione nei percorsi di formazione degli insegnanti. La circostanza meriterebbe uno studio specifico, ma noi crediamo che il ritardo nell'instaurare un confronto e un dialogo approfonditi, volti a conciliare le sensibilità di orientamento pedagogico con quelle di orientamento scientifico, abbiano determinato una certa resistenza nel riconoscere il potenziale culturale ed educativo dell'informatica e, appunto, dell'attività di programmazione.

1. La confusione assai diffusa fra *uso* delle applicazioni informatiche e l'impresa *concettuale* di immaginarne l'utilità, progettarle, realizzarle, convalidarne il funzionamento e comprenderne l'impatto. Infatti, si è posta un'enfasi eccessiva sulla necessità di acquisire abilità di utilizzo di applicazioni specifiche, trascurando quasi completamente i principi scientifici che le hanno rese possibili. Un po' come se fosse sufficiente assaggiare molte meringhe con la panna per apprendere quanto la fisica, la chimica, o semplicemente la gastronomia siano cruciali in cucina: da questa enfasi mal riposta nemmeno i "pasticcieri" potrebbero trarre un beneficio significativo.
2. L'equivoco rapporto con discipline strutturate più stabilmente nella nostra tradizione scolastica, in particolare la matematica con la quale l'informatica ha spesso una relazione edipica (dando quasi l'impressione di volerla "superare" uccidendola), mentre sarebbe più proficuo se coltivasse le proprie potenzialità di nuovo strumento di pensiero, complementare e non alternativo ad altri più consolidati.
3. La percezione della programmazione come attività fortemente specialistica, utile solo a chi intende sviluppare una professionalità in settori tecnologici. Questa visione limitata trascura il fatto che l'elaborazione automatica di informazioni ha caratteristiche molto peculiari, ed è capace di influire grandemente sulla nostra consapevolezza del mondo, consapevolezza che è essa stessa basata su una elaborazione di informazioni — per definizione non automatica.

Le distorsioni a cui si è accennato sono emerse per motivi comprensibili, ma hanno avuto e tuttora hanno conseguenze negative. Per questo pensiamo possa essere utile mettere in luce, in una forma auspicabilmente accessibile a tutte le parti coinvolte in progetti educativi, gli aspetti centrali della programmazione, cioè *di che cosa parliamo* quando (noi informatici) sosteniamo che è importante fare l'esperienza della programmazione per capire alcuni aspetti del mondo che ci circonda senza subirne acriticamente l'impatto.

A tal fine abbiamo formato un gruppo di lavoro [23] che si è riunito a partire da marzo 2022 e che comprende studiosi con una lunga esperienza non solo nella didattica dell'informatica, ma anche nella storia e nella filosofia della disciplina, nonché nella diffusione delle tecnologie digitali. Ne è risultato un documento [24], che qui intendiamo riassumere, rivolto principalmente agli insegnanti e agli altri

attori coinvolti nelle politiche scolastiche. L'obiettivo è fornire uno strumento che consenta di sviluppare una visione più chiara e articolata della *natura* dei programmi [2] e delle attività connesse alla loro realizzazione, e che inoltre possa fungere da guida ai fini della scelta dei temi più rilevanti per garantire l'efficacia delle azioni educative e pedagogiche.

2 Caratterizzazione dei programmi nella letteratura

A prima vista potrebbe sembrare facile definire *che cos'è un programma*: in molti manuali, per esempio, è introdotto semplicemente come una sequenza di istruzioni per un *computer*. E in effetti termini come *coding*, molto usato anche nei documenti del Ministero dell'Istruzione, trasmettono di fatto questa idea riduttiva [25, 4]. Ma in realtà si tratta di un concetto molto più sfaccettato e ricco, come si può evincere da una lettura attenta della storia dell'informatica e da un'analisi di come l'elaborazione automatica di informazioni descritta dai programmi sia ormai inestricabilmente legata alla maggior parte delle nostre attività quotidiane, dalla guida delle automobili all'espressione artistica. È ormai difficile, infatti, trovare ambiti rispetto ai quali la commistione con la programmazione non abbia influito in modi anche radicali.

Alla luce della centralità che i programmi e la programmazione assumono nell'informatica, ci si dovrebbe aspettare che esista una definizione chiara e condivisa di che cos'è un programma. In realtà, a partire degli esordi della disciplina, se ne possono trovare svariate. Per esempio, una definizione influente da parte di Grace Hopper apparve nel *First glossary of programming terminology* pubblicato dall'ACM nel 1954 [14], definizione in seguito ripresa in diversi glossari:

Programma: un piano per risolvere un problema. Un programma completo comprende i piani per la trascrizione dei dati, la codifica per il computer e i piani per l'acquisizione dei risultati nel sistema. La lista delle istruzioni codificate è denominata **routine**.

Una caratterizzazione in termini di pianificazione sequenziale di istruzioni codificate, introdotta nell'ambito del progetto ENIAC (uno dei primi elaboratori digitali), ispira anche quelle prevalenti oggi [5]; in particolare, fu adottata in relazione ai primi linguaggi di programmazione di alto-livello come ALGOL:

Sequenze di istruzioni e dichiarazioni, combinate opportunamente, sono chiamate programmi. [26]

Negli anni '50 e nei primi anni '60 del Novecento i campi di applicazione della programmazione erano ancora piuttosto circoscritti, orientati prevalentemente al calcolo scientifico e aziendale, e l'odierna ubiquità dei programmi era inimmaginabile per i costi dell'hardware, accessibili solo a grandi organizzazioni. Malgrado il modo in cui oggi si sviluppano e si utilizzano i programmi si sia certamente evoluto rispetto agli albori dell'informatica, alcune concezioni che risalgono a quell'epoca restano ancora molto popolari fra gli informatici.

La natura dei programmi assume inoltre un ruolo chiave nel contesto più ampio della discussione riguardo alla natura dell'informatica, a partire da alcuni contributi significativi che sono apparsi negli anni '80 e '90 (per esempio [20, 7, 29]), discussione stimolata in particolare dalla domanda se l'informatica — computer *science* nel mondo anglosassone — sia effettivamente una scienza. Contributi più recenti al riguardo sono riportati in varie fonti, fra cui si possono segnalare [6, 9, 27, 17, 28, 30, 1]. Da questo dibattito emerge che l'informatica non ha un'identità chiara e tende piuttosto a delinearci come un'area interdisciplinare.

I diversi modi di concepire i programmi e la programmazione spesso ne enfatizzano solo un aspetto particolare. Il progettista di un linguaggio di programmazione, per esempio, tenderà principalmente a guardare i programmi dal punto di vista della notazione e della semantica formale; un informatico teorico spesso li considera come procedure astratte; un ingegnere del software, d'altro canto, sarà più interessato alla complessità dei programmi e alle rispettive relazioni con il mondo reale. Donald Knuth, per esempio, ha inteso i programmi innanzitutto come implementazioni di algoritmi [19], ma è interessante notare che più tardi lo stesso Knuth prefigura una caratterizzazione abbastanza diversa, assimilando i programmi a “opere letterarie”, da cui l'approccio che oggi è conosciuto come *literate programming*:

Invece di immaginare che il nostro compito principale sia di istruire un computer su quello che deve fare, [dovremmo] piuttosto concentrarci [sul compito di] spiegare agli esseri umani quello che vogliamo che il computer faccia. [21]

Edsger Dijkstra, come altri studiosi, si è concentrato sulla natura matematica dei programmi [10]. In polemica con questo punto di vista, il filosofo James Fetzer enfatizzò la distinzione fra programmi come entità matematiche *astratte* e programmi come modelli *causali* in quanto oggetti fisici:

Da un punto di vista metodologico, si può dire che i programmi sono congetture, mentre le esecuzioni sono tentativi di refutazioni — che hanno fin troppo spesso successo. [13]

Secondo la prospettiva ingegneristica i programmi sono componenti di sistemi più complessi [8], e quindi non sono riducibili alla loro notazione o a una struttura matematica. Per Jackson [16], in particolare, la terna *requisiti – specifica – programma* è centrale al fine di mettere in relazione la macchina e il mondo. Altri autori hanno proposto tassonomie a supporto di una comprensione più diversificata dei programmi. Un esempio ben conosciuto, sempre nell'ambito dell'ingegneria del software, è il lavoro di Lehman sulle leggi alla base dell'evoluzione dei programmi [22], dove i programmi sono classificati in termini di complessità e “intreccio” con il mondo.

Definizioni e classificazioni sono questioni tradizionali anche per la filosofia analitica. Un'idea chiave alla base della caratterizzazione di Colburn [6] fa riferimento al ruolo peculiare dell'astrazione in informatica e nella programmazione. Nel saggio di Turner [30] i programmi sono invece analizzati

attraverso la lente della filosofia degli artefatti tecnologici, basata sulla dicotomia struttura/funzione, e del processo di progettazione. Più specificamente, per Turner i programmi sono definiti dalla terna *specificità - programma simbolico - processo fisico*. Eden [11], d'altro canto, discute le contrastanti caratterizzazioni ontologiche, metodologiche ed epistemologiche dei programmi secondo le diverse prospettive ispirate alla matematica, all'ingegneria e alla scienza. Per portare un ulteriore esempio, Irmak [15] confronta invece i programmi con la musica, osservando che, analogamente alla musica, i programmi hanno una natura duale. L'idea della natura duale dei programmi è comunque ricorrente; come osservato da Alan Kay:

Un messaggio intangibile incorporato in un supporto materiale è l'essenza del software [...]. Il computer è un veicolo da guidare o un saggio da scrivere? Gran parte della confusione deriva dal tentativo di risolvere la questione a questo livello. La natura proteiforme del computer sta nel fatto che può comportarsi come una macchina o come un linguaggio a cui dare forma per trarne beneficio. [18]

Le caratterizzazioni esemplificate mostrano come una certa concezione dei programmi di solito tenda a confermare una particolare visione dell'informatica. Di conseguenza, proprio come per l'informatica in generale, ci ritroviamo di fronte a un ventaglio variegato di definizioni, ciascuna delle quali getta luce su un aspetto diverso. Non deve quindi sorprendere che anche nella didattica dell'informatica si ritrovino diverse prospettive su come i programmi debbano essere insegnati e con quali obiettivi. Forse ciò spiega in parte la proliferazione di termini quali *coding*, *computational thinking* e *programmazione*.

Recentemente la natura poliedrica dei programmi è stata riconosciuta in diversi contesti. Il lavoro collaborativo [2] prende le mosse proprio a partire dall'osservazione che i "programmi" sono oggetti complessi che comprendono molti aspetti: sono reali — hanno effetti sulla nostra vita; sono astratti — elaborano entità astratte; sono concreti — occupano uno spazio nella memoria dei dispositivi digitali e possono essere copiati, trasferiti o corrotti. Il progetto PROGRAMme, che si rivolge a storici, filosofi e studiosi di informatica, propone un'analisi dei programmi in termini delle loro diverse modalità: fisica, socio-tecnica e notazionale (<https://programme.hypotheses.org/>). Si può quindi concludere che un approccio olistico ai programmi sia il più adatto al fine di garantire una "alfabetizzazione" *generale*, ovvero, con le parole di Alan Kay:

Indirizzare la magia del mezzo verso il perseguimento delle proprie finalità, anziché contro di esse, significa acquisire l'alfabetizzazione. [18]

3 Le sei "facce" di un programma

Attraverso le discussioni maturate in seno al gruppo di lavoro [23], alla luce delle prospettive maturate negli ambiti filosofico e didattico disciplinare, abbiamo identificato sei aspetti chiave che, a nostro avviso, sono i più salienti per caratterizzare la natura dei programmi e per spiegarne l'impatto rivoluzionario nella società. Non intendiamo certamente avanzare alcuna pretesa di esaustività:

anche la visione d'insieme qui proposta è probabilmente riduttiva da certi punti di vista. Tuttavia, ci sembra sufficiente a motivare l'interesse che la Scuola dovrebbe dedicare all'argomento.

In sintesi, la caratterizzazione della natura dei programmi può essere così schematizzata:

1. I programmi sono *strumenti* utilizzabili;
2. I programmi sono artefatti tecnologici *opera dell'uomo*;
3. I programmi sono *oggetti fisici*;
4. I programmi sono *entità astratte*;
5. I programmi sono *eseguibili automaticamente*;
6. I programmi sono *artefatti linguistico-notazionali*.

Le sei "facce" così introdotte sono sintetizzate iconicamente nella figura 1, che inoltre presenta ai lati opposti di un esagono coppie di aspetti *duali*: essi caratterizzano la natura dei programmi proprio nella tensione dialettica che li contrappone.

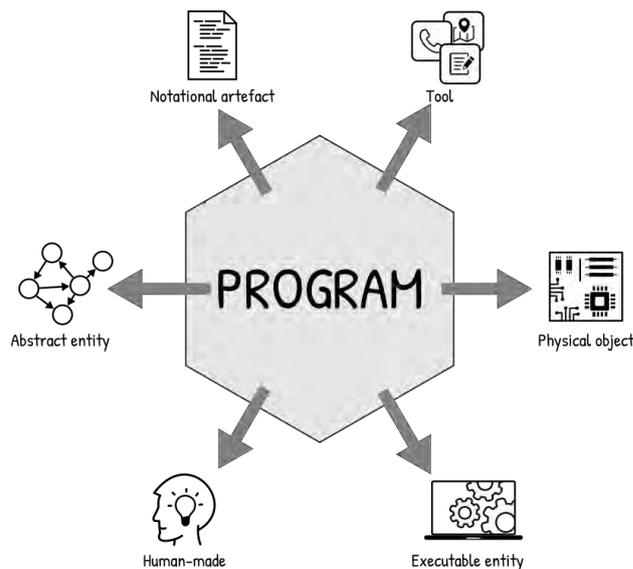


Figura 1

I sei aspetti chiave della natura di un programma (figura tratta dal documento originale [24])

3.1 Programmi come strumenti

Questo è forse l'aspetto più evidente per molte persone. I programmi sono gli strumenti attraverso i quali svolgiamo moltissime delle nostre attività. In larga misura la società dipende dalla disponibilità di questi strumenti e i programmi ci permettono di affrontare compiti che altrimenti sarebbero al di fuori delle nostre possibilità individuali, anche amplificando le nostre capacità cognitive. Grazie ai programmi lo stesso dispositivo fisico (p. es. uno *smartphone*) può assolvere molte funzioni diversissime fra loro: dalla contabilità aziendale, all'ascolto della

musica, dalla fotografia al montaggio cinematografico, e così via. Tuttavia, è bene tenere presente che uno strumento non è mai neutrale, ma incorpora scelte e valori impliciti: occorre tener conto di questo anche quando la nostra esperienza è mediata da un programma — benché sia facile dimenticarsene a causa dell'ubiquità, della flessibilità e dell'invisibilità degli strumenti *software*.

3.2 Programmi come opera dell'uomo

I programmi sono un'opera umana, realizzati intenzionalmente per soddisfare qualche esigenza umana, con uno spettro molto ampio: si può scrivere un programma per risolvere un problema matematico, o portare a termine una ricerca in una banca dati, ma anche per soddisfare un impulso artistico o semplicemente per... imparare a programmare. In ogni caso si tratta di qualcosa costruito dall'uomo con uno scopo, che non sempre è allineato con gli scopi di chi il programma poi lo usa: chi sviluppa programmi che permettono la formazione di *social network* ha in genere obiettivi piuttosto diversi (p. es. raccogliere informazioni utili per il mercato pubblicitario) rispetto a chi usa tali programmi per condividere informazioni multimediali. La realizzazione di un programma richiede spesso un'attività complessa, svolta cooperativamente da gruppi di lavoro eterogenei che devono rispondere a moltissimi interessi e pulsioni, oltre a rispettare i vincoli tecnologici.

3.3 Programmi come oggetti fisici

I programmi sono anche oggetti fisici, un aspetto che è facile sottovalutare, ma che ha invece una grande rilevanza. Il programma che stiamo usando deve risiedere da qualche parte per poter essere eseguito da una macchina. È potenzialmente soggetto ad alterazioni, volontarie e involontarie, che possono impedirne il funzionamento oppure determinare effetti inaspettati/indesiderati. Inoltre, l'esecuzione di programmi e talvolta anche la semplice conservazione consumano energia.

3.4 Programmi come entità astratte

I programmi sono poi entità astratte: rappresentano manipolazioni di simboli, cioè di segni privi di significato intrinseco, cosicché il significato prende forma esclusivamente nella mente di chi li concepisce o ne prefigura il funzionamento. L'interprete (meccanico/elettronico), nel dare corso all'elaborazione descritta dal programma, non fa altro che operare in accordo con i suoi principi costruttivi, facendo evolvere lo stato dei suoi componenti in maniera prevedibile. L'interesse dell'elaborazione scaturisce dunque dalle proprietà astratte, algoritmiche dei programmi, che occorre imparare ad apprezzare e a mettere in relazione con la realtà concreta, valutandone le semplificazioni implicate e le potenziali alternative, oltre che la correttezza rispetto ai risultati attesi.

3.5 Programmi come entità eseguibili automaticamente

I programmi vengono eseguiti automaticamente, in contesti in cui non è previsto alcun intervento da parte dei progettisti, e ciò ne costituisce una caratteristica fondamentale. In altri termini, il programmatore deve immaginare ciò che accadrà o che potrebbe accadere durante l'esecuzione, ma una volta realizzato,

l'esecuzione del programma è fuori dal suo controllo: per cambiare qualcosa serve produrre una nuova versione. L'elaborazione dei dati di input (manuali o provenienti da dispositivi collegati e sensori) avviene nella maniera cieca e meccanica descritta al punto precedente; qualsiasi valutazione di ragionevolezza o affidabilità è responsabilità di coloro che hanno concorso allo sviluppo del sistema.

3.6 Programmi come artefatti linguistico-notazionali

I programmi sono inoltre uno strumento per comunicare idee e strategie non solo agli interpreti che li eseguono passivamente, ma anche ad altre persone che possono analizzarli, apprezzandone tutti i dettagli grazie alla precisione imposta dalla possibilità dell'esecuzione automatica. La varietà dei linguaggi di programmazione dimostra un'inesauribile ricerca di modalità espressive che si prestino alla modellazione e alla risoluzione dei problemi che si vogliono affrontare. Nello stesso tempo, le notazioni utilizzate per rappresentare l'informazione tendono intrinsecamente a condizionare il modo in cui riusciamo a ragionare per mezzo di esse.

4 L'impresa concettuale delle elaborazioni automatiche

I programmi si scrivono per esprimere soluzioni automatiche di problemi computazionali e quindi assumono significato in relazione alle varie entità schematizzate nella mappa concettuale illustrata in figura 2.

I problemi computazionali sono il corrispettivo in termini di elaborazione dell'informazione (ovvero, nel caso digitale, di manipolazione di rappresentazioni simboliche) di un'esigenza nel mondo reale. I problemi computazionali modellano



Figura 2

Una mappa concettuale che illustra l'attività della programmazione nel suo complesso (figura tradotta dal documento originale [24])

i bisogni nel mondo reale rappresentando le informazioni rilevanti con dati opportuni. Un programma può essere eseguito automaticamente — in altre parole un sistema fisico (per esempio un computer o uno smartphone) può dare corso all'elaborazione di informazioni descritte dal programma. Normalmente questi sistemi sono in grado di trattare solo rappresentazioni digitali, cioè simboliche, dei dati. I dati, perciò, devono essere codificati tramite sequenze di simboli tratti da un alfabeto predefinito.

Un algoritmo è l'idea astratta di un metodo (ossia una procedura precisa che può essere portata a termine seguendola in modo *prescrittivo*) per risolvere un problema computazionale. L'algoritmo viene progettato pensandone l'esecuzione da parte di un agente computazionale idealizzato, capace di eseguire un insieme ridotto di azioni primitive. Un sistema di calcolo reale (potrebbe essere un computer o un altro dispositivo hardware, ma anche un altro programma, come l'interprete di un linguaggio di programmazione) è un esemplare specifico di tale agente computazionale idealizzato. Affinché possa essere eseguito automaticamente da un sistema di calcolo reale, un algoritmo deve essere implementato, cioè codificato in un linguaggio di programmazione rispettandone le rigide regole necessarie a evitare ambiguità e adattandolo ai vincoli tecnologici specifici del sistema: solo così un algoritmo si traduce effettivamente in un programma.

4.1 Un esempio

La figura 3 illustra i concetti che appaiono nella mappa di figura 2 facendo riferimento a un programma (quello della “cassa del supermercato”) progettato per gestire la produzione degli scontrini di un negozio. I prezzi degli articoli



Figura 3

Un esempio: il programma della cassa del supermercato (figura tradotta dal documento originale [24])

venduti possono essere modellati come numeri. Perciò, l'elaborazione di uno scontrino corrisponde al generico problema computazionale di "sommare una serie di numeri". Il programma della cassa del supermercato è eseguito da un registratore di cassa, un dispositivo in grado di acquisire i prezzi degli articoli in vendita (digitati da un operatore o letti interpretando un codice a barre), quindi di stampare gli scontrini corrispondenti. Il dispositivo, in realtà, manipola i numeri rappresentati tramite sequenze di simboli binari — basati cioè su due stati chiaramente distinguibili dei suoi componenti elettronici.

L'algoritmo tipico per sommare una serie di numeri è il seguente:

1. usa una variabile per tener traccia del totale, assegnandovi inizialmente il valore zero;
2. fintantoché la lista dei numeri non è esaurita, ripeti l'istruzione seguente:
 - 2.1. somma il prossimo numero alla variabile che tiene traccia del totale;
3. stampa il valore della variabile che tiene traccia del totale e termina.

Durante la progettazione di questo algoritmo, assumiamo che l'esecutore sia in grado di ricevere in input qualsiasi numero, sommare due numeri e stampare qualsiasi numero. Affinché il registratore di cassa possa eseguire *fisicamente* queste operazioni occorre tener conto di alcuni vincoli specifici, per esempio i limiti ai valori ed eventualmente alla quantità di numeri oggetto dell'elaborazione. L'algoritmo probabilmente viene implementato in un linguaggio di programmazione specializzato per i registratori di cassa. Sia il problema computazionale che l'algoritmo, però, sono molto generali e possono essere facilmente adattati ad altri scopi analoghi, come per esempio calcolare la popolazione totale di un Paese sulla base delle popolazioni delle regioni che lo costituiscono. Perciò, un'implementazione in un linguaggio di programmazione non specifico permetterebbe di usare lo stesso programma, magari con qualche piccolo aggiustamento, anche in contesti piuttosto diversi.

5 Conclusione

Lo scopo del gruppo di lavoro è fornire una chiara esposizione degli aspetti chiave dei programmi che possa aiutare gli insegnanti e altri operatori culturali a identificare le ragioni della centralità del software nella società attuale e a orientare al meglio gli interventi educativi. La versione italiana preliminare del documento è disponibile all'indirizzo <https://aladdin.unimi.it/naturadeiprogrammi.pdf>. La versione definitiva è attesa per la fine del 2022 e crediamo che possa diventare una risorsa importante ai fini di progettare percorsi formativi in cui la pratica della programmazione e la riflessione sui programmi vengano intesi innanzitutto come strumenti di consapevolezza e cittadinanza attiva.

Riferimenti bibliografici

[1] Tim Bell, Paul Tymann, and Amiram Yehudai. The Big Ideas in Computer Science for K–12 Curricula. Bull. EATCS 124, 11 pages, 2018.

[2] Andrej Brodnik, Andrew Csizmadia, Gerald Futschek, Lidija Kralj, Violetta Lonati, Peter Micheuz, and Mattia Monga. Programming for All: Understanding the Nature of Programs. CoRR abs/2111.04887, 2021, 25 pages. arXiv:2111.04887

[3] Francesca Caena and Christine Redecker. Aligning teacher competence frameworks to 21st century challenges: The case for the European Digital Competence Framework for Educators (Digcompedu). European Journal of Education, 2019, 54. <https://doi.org/10.1111/ejed.12345>.

[4] Isabella Corradini, Michael Lodi, and Enrico Nardelli. An investigation of italian primary school teachers' view on coding and programming. In Proc. of Informatics in Schools. Fundamentals of Computer Science and Software Engineering – ISSEP 2018. Springer International Publishing, 2018, pp. 228–243, S.N. Pozdniakov and V. Dagienė (Eds.), Cham, Switzerland.

[5] Liesbeth De Mol and Maarten Bullynck. Roots of 'program' Revisited. Commun. ACM, 64(4), 2021, pp. 35–37. <https://doi.org/10.1145/3419406>

[6] Timothy Colburn. Philosophy and Computer Science. Routledge (Taylor & Francis), 1999, New York, NY.

[7] Peter J. Denning, Douglas E. Comer, David Gries, Michael C. Mulder, Allen Tucker, A. Joe Turner, and Paul R Young. Computing as a discipline. Computer, 22(2), 1989, pp. 63–70.

[8] Peter J. Denning. The Field of Programmers Myth. Commun. ACM, 47 (7), 2004, pp. 15–20. <https://doi.org/10.1145/1005817.1005836>

[9] Peter J. Denning and Craig H. Martell. Great Principles of Computing. The MIT Press, 2015, Cambridge, MA.

[10] Edsger W. Dijkstra. On the cruelty of really teaching computer science. Personal communication, 1988.

<https://www.cs.utexas.edu/users/EWD/ewd10xx/EWD1036.PDF>

[11] Amnon H. Eden. Three Paradigms of Computer Science. Minds Mach, 17 (27), 2007, pp. 135–167. <https://doi.org/10.1007/s11023-007-9060-8>

[12] Katrina Falkner, Sue Sentance, Rebecca Vivian, Sarah Barksdale, Leonard Busuttil, Elizabeth Cole, Christine Liebe, Francesco Maiorana, Monica M. McGill, and Keith Quille. An international comparison of K–12 computer science education intended and enacted curricula. In Proceedings of the 19th Koli Calling International Conference on Computing Education Research, Koli Calling '19, New York, USA, 2019. Association for Computing Machinery.

[13] James H. Fetzer. Program Verification: The Very Idea. Commun. ACM, 31 (9), 1988, pp. 1048–1063. <https://doi.org/10.1145/48529.48530>

[14] Grace M. Hopper, chairman. ACM first glossary of programming terminology. Report to the ACM from the Committee on Nomenclature, 1954.

[15] Nurbay Irmak. Software is an Abstract Artifact. Grazer Philosophische Studien, 86(1), 2012, pp. 55–72.

[16] Michael Jackson. The World and the Machine. In Proceedings of the 17th International Conference on Software Engineering – ICSE '95, Seattle, USA, 1995. Association for Computing Machinery, New York, NY, USA, pp. 283–292. <https://doi.org/10.1145/225014.225041>

- [17] Lars-Erik Janlert. The program is the solution — what is the problem? European Conference on Computing and Philosophy, 2006.
- [18] Alan Kay. Computer software. *Scientific American*, 251, 1984, pp. 53–59.
- [19] Donald E. Knuth. *The Art of Computer Programming (1st Edition) Vol. 1*. Addison-Wesley, 1968, Boston, USA.
- [20] Donald E. Knuth. Computer Science and its Relation to Mathematics. *The American Mathematical Monthly*, 81(4). 1974, pp. 323–343.
<https://doi.org/10.1080/00029890.1974.11993556>
- [21] Donald E. Knuth. Literate Programming. *Comput. J.*, 27(2), 1984, pp. 97–111. <https://doi.org/10.1093/comjnl/27.2.97>
- [22] Meir M. Lehman. Programs, life cycles, and laws of software evolution. In *Proc. IEEE*, 68(9), 1980, pp. 1060–1076.
<https://doi.org/10.1109/PROC.1980.11805>
- [23] Violetta Lonati, Andrej Brodnik, Tim Bell, Andrew Paul Csizmadia, Liesbeth De Mol, Henry Hickman, Therese Keane, Claudio Mirolo, Mattia Monga, and Matti Tedre. Characterizing the nature of programs for educational purposes. In *Proceedings of the 27th ACM Conference on Innovation and Technology in Computer Science Education Vol. 2, ITICSE '22*, pp. 572–573, New York, NY, USA, 2022. Association for Computing Machinery.
- [24] Violetta Lonati, Andrej Brodnik, Tim Bell, Andrew Paul Csizmadia, Liesbeth De Mol, Henry Hickman, Therese Keane, Claudio Mirolo, and Mattia Monga. The nature of programs, or: What we talk about when we talk about programs. <https://drive.google.com/file/d/1hVPDhSHu3ivRXvcK7BVEhSmpqz5izow> (ultimo accesso, novembre 2022).
- [25] Violetta Lonati, Dario Malchiodi, Mattia Monga, and Anna Morpurgo. Is coding the way to go? In *Proc. of the 8th International Conference on Informatics in Schools: Situation, Evolution, and Perspective – ISSEP 2015*. LNCS, Vol. 9378, pp. 165–174, 2015. Springer International Publishing, A. Brodnik and J. Vahrenhold (Eds.).
- [26] Alan J. Perlis and Klaus Samelson. Preliminary report – International algebraic language. *Commun. ACM*, 1(12), 1958, pp. 8–22.
http://www.softwarepreservation.org/projects/ALGOL/report/Algol58_preliminary_report_CACM.pdf
- [27] William J. Rapaport. Philosophy of Computer Science: An Introductory Course. *Teaching Philosophy*, 4(28), 2005, pp. 319–341.
- [28] Matti Tedre. *The Science of Computing: Shaping a Discipline*. CRC Press, 2014, Boca Raton, FL.
- [29] Sherry Turkle and Seymour Papert. Epistemological pluralism: Styles and voices within the computer culture. *Signs: Journal of women in culture and society*, 16(1), 1990, pp. 128–157.
- [30] Raymond Turner. *Computational Artifacts: Towards a Philosophy of Computer Science*. Springer, 2018, Cham, Switzerland.

Sviluppo del pensiero computazionale nella scuola dell'infanzia e nella scuola primaria

Alice Barana¹, Francesca Casasso¹, Cecilia Fissore², Valeria Fradiante¹, Marina Marchisio¹

¹Dipartimento di Biotecnologie Molecolari e Scienze per la Salute, Università di Torino, Via Nizza, 52, 10126 Torino TO, Italia

²Dipartimento di Lingue e Letterature Straniere e Culture Moderne, Università di Torino, Via Giuseppe Verdi Fronte 41, 10124 Torino TO, Italia

{alice.barana, francesca.casasso, cecilia.fissore, valeria.fradiante, marina.marchisio}@unito.it

1

0

Sommario

L'importanza di sviluppare competenze di pensiero computazionale è sottolineata fin dalla scuola dell'infanzia, al fine di insegnare agli alunni come risolvere i problemi in modo analitico, scomporli in sotto problemi e pianificare la soluzione più adatta. È quindi necessario formare gli insegnanti per aiutare gli alunni a sviluppare queste competenze. Il contesto di questa ricerca è un corso di formazione online di 14 ore con 21 insegnanti della scuola dell'infanzia e primaria della Valle d'Aosta. Dopo aver presentato i concetti teorici e alcuni esempi di attività laboratoriali, è stato chiesto agli insegnanti di progettare, seguendo un'apposita scheda, due attività e di sperimentarne una con i propri alunni. Alla fine del percorso di formazione gli insegnanti hanno compilato due questionari: uno di gradimento e uno per la spiegazione della sperimentazione e dei risultati ottenuti in classe. La domanda di ricerca di questo articolo è: quale tipologia di attività può aiutare gli alunni a sviluppare competenze di pensiero computazionale? Quali processi mentali vengono attivati durante queste attività? Per rispondere a questa domanda sono state analizzate le attività progettate e sperimentate dagli insegnanti nelle loro classi e le risposte degli insegnanti ai due questionari. Le attività sono state efficaci nello sviluppo del pensiero critico e delle competenze di pensiero computazionale. I processi mentali più stimolati dalle attività progettate e sperimentate dagli insegnanti sono stati il pensiero algoritmico, il pensiero logico e la decomposizione. Gli insegnanti hanno apprezzato molto il corso e le metodologie proposte, e ritengono che lo sviluppo di queste competenze sia molto importante anche nell'ambito di un curriculum verticale.

1

0

1 Introduzione

La ricerca degli ultimi decenni ha dimostrato l'importanza di sviluppare in ambito educativo competenze di problem solving e pensiero computazionale, competenze fondamentali nell'era digitale proprio come leggere, scrivere e contare. Questo viene sottolineato sia nelle Indicazioni Nazionali per la scuola dell'infanzia e per il primo ciclo di istruzione (MIUR, 2012) sia nei quadri di riferimento europei, come ad esempio il DigComp (Quadro di riferimento per le competenze digitali dei cittadini) (Commissione Europea, 2016). Lo sviluppo di competenze di pensiero computazionale e di problem solving gioca un ruolo importante nell'educazione, fin dalla scuola dell'infanzia in una prospettiva di curriculum verticale. Gli insegnanti possono utilizzare diverse tipologie di attività (disponibili in rete o nei manuali scolastici) o possono progettare loro stessi delle attività per sviluppare questo tipo di competenze. In Italia, il Ministero dell'Istruzione promuove da diversi anni progetti per l'introduzione di metodologie innovative volte allo sviluppo del pensiero computazionale e delle competenze di problem solving, anche attraverso l'uso delle tecnologie in aula e la loro integrazione con le risorse tradizionali. Uno di questi è il progetto "PP&S" (www.progettopp.it), Problem Posing and Solving, che promuove la formazione dei docenti provenienti da tutta Italia di ogni grado e disciplina (Barana et al., 2020; Fissore & Marchisio, 2020). Il contesto di questa ricerca, analizzato più nel dettaglio in seguito, è il corso di formazione dal titolo "Percorso di formazione per gli insegnanti: progettare e valutare attività per lo sviluppo di competenze di problem solving e pensiero computazionale", che ha coinvolto 21 insegnanti, di cui 6 della scuola dell'infanzia e 15 della scuola primaria, tutti appartenenti all'Istituto "O. Jacquemet" di Verrès, in Valle d'Aosta¹. Inizialmente, il corso di formazione doveva tenersi in presenza ma a causa della pandemia di COVID-19, il percorso formativo si è svolto online, rimodulando le attività previste e la durata degli incontri in modo da mantenere un carattere interattivo e coinvolgente. Dopo la presentazione dei concetti teorici e di alcuni esempi di attività laboratoriali per lo sviluppo delle competenze di problem solving e di pensiero computazionale, è stato chiesto agli insegnanti di progettare due attività: una per lo sviluppo delle competenze di pensiero computazionale e una per lo sviluppo delle competenze di problem solving. Gli insegnanti hanno poi scelto una delle due attività e l'hanno sperimentata in classe con i loro alunni. Dopo la sperimentazione, hanno risposto ad un questionario relativo alle attività sperimentate per spiegare e commentare la loro esperienza e illustrare i risultati ottenuti. Nella documentazione hanno incluso anche alcune testimonianze delle attività, come fotografie, documenti e altro materiale didattico di supporto allo svolgimento dell'attività. All'inizio del corso tutti gli insegnanti hanno compilato un questionario iniziale per sondare le loro esperienze pregresse e alla fine del corso un questionario di gradimento. Le

¹ Le autrici ringraziano la dirigente Antonella Dallou per aver promosso il corso di formazione e tutti i docenti per aver partecipato con entusiasmo.

domande di ricerca di questo articolo sono: quale tipologia di attività può aiutare gli alunni a sviluppare competenze di pensiero computazionale? Quali processi mentali vengono attivati durante queste attività?

2 Stato dell'Arte

Nel 1980 Papert (Papert, 1980) ha introdotto per la prima volta il termine "pensiero computazionale", senza dare una definizione specifica. La sua idea di pensiero computazionale era legata alla programmazione informatica, che secondo lui rappresenta uno strumento utile per gli studenti per rendere più concreti i concetti astratti e per costruire modelli mentali di ciò che stanno imparando (Lodi et al., 2020). Traendo spunto da questo lavoro, Jeanette Wing nel 2006 (Wing, 2006) ha fornito una prima caratterizzazione di pensiero computazionale come forma di pensiero che non è vincolata all'aspetto tecnologico ma che è possibile ritrovare in diversi aspetti del vivere umano. In particolare, l'autrice ha descritto il pensiero computazionale come "i processi mentali coinvolti nella formulazione di un problema e nell'esprimerne la/e soluzione/i in modo tale che un computer, umano o macchina, possa effettivamente svolgerli". In questo caso, il pensiero computazionale è descritto come "un'abilità fondamentale per tutti, non solo per gli informatici", importante quanto la lettura, la scrittura e l'aritmetica, e che quindi dovrebbe essere aggiunta alle competenze di base del curriculum (Wing, 2006, 2017). Il pensiero computazionale è strettamente correlato al problem solving e può essere inteso come un processo in cui è possibile individuare le seguenti fasi (Kalelioğlu et al., 2016):

- identificazione di un problema;
- raccolta/rappresentazione/analisi di dati;
- generazione/selezione/progettazione di una soluzione;
- implementazione della soluzione;
- valutazione/miglioramento della soluzione.

In tale modello, ciascuna fase individua un insieme differente di processi cognitivi. Ad esempio, l'identificazione di un problema implica processi di astrazione e decomposizione. Mentre l'astrazione consente di rimuovere gli aspetti irrilevanti di un problema in modo da concentrarsi sulle componenti cruciali, la decomposizione consente di suddividere problemi complessi in problemi più piccoli per proporre soluzioni adeguate. La raccolta dei dati e la loro analisi permettono di comprendere un problema attraverso, ad esempio, il riconoscimento di pattern. La generazione delle soluzioni richiede l'abilità di pensiero algoritmico (Città et al., 2021). Anche Lodi e colleghi (Lodi et al., 2017) descrivono il pensiero computazionale come un processo mentale (o più in generale un modo di pensare) per risolvere problemi (problem solving) e ne definiscono gli elementi costitutivi: strategie mentali, metodi, pratiche e competenze trasversali. Per quanto riguarda le strategie mentali utili per risolvere problemi, gli autori concettualizzano i seguenti processi mentali:

- pensiero algoritmico: usare il pensiero algoritmico per progettare una sequenza ordinata di passi (istruzioni) per risolvere un problema;
- pensiero logico: usare la logica e il ragionamento per stabilire e controllare i fatti;
- scomposizione di problemi: dividere e modularizzare un problema complesso in semplici sotto- problemi, risolubili in modo più semplice;
- astrazione: liberarsi dei dettagli inutili per concentrarsi sulle idee rilevanti;
- riconoscimento di pattern: individuare regolarità/schemi ricorrenti nei dati e nei problemi;
- generalizzazione: usare le regolarità riconosciute per fare previsioni o per risolvere problemi più generali.

Ciò che distingue il pensiero computazionale dal problem solving è il cambio di paradigma concettuale costituito dal passaggio dal risolvere problemi al fare problem solving (Lodi et al., 2017). Il pensiero computazionale non riguarda solo la risoluzione di un problema, include un passaggio in più: la formulazione del problema e la soluzione devono essere espresse scrivendo un algoritmo in un linguaggio appropriato, in modo che un "agente di elaborazione delle informazioni" (umano o macchina) possa comprendere, interpretare ed eseguire le istruzioni fornite. Se immaginiamo un'attività per stimolare la competenza di pensiero computazionale degli alunni, ci aspettiamo una situazione problematica la cui soluzione deve essere espressa attraverso un algoritmo in un linguaggio specifico. Qualcuno (un essere umano o una macchina) interpreta il ruolo dell'agente che elabora le informazioni ed esegue le istruzioni fornite. Anche se l'utilizzo delle tecnologie può supportare lo sviluppo di competenze di pensiero computazionale (Barana et al., 2019), non è necessario il loro utilizzo per proporre in classe attività per lo sviluppo questa competenza.

Un aspetto importante nella progettazione di attività di pensiero computazionale e non solo, risulta essere il contesto delle situazioni problematiche. Un contesto ludico e molto vicino agli alunni può aiutare a stimolare il loro interesse e coinvolgimento. Il pensiero computazionale e il problem solving sono competenze diverse ma correlate, ed è fondamentale stimolarle entrambe anche in un contesto educativo fin dalla scuola dell'infanzia, sfruttando anche le situazioni ludiche (MIUR, 2012).

3 Contesto della ricerca

L'obiettivo del percorso di formazione è stato la progettazione di attività per lo sviluppo di competenze di problem solving e di pensiero computazionale. Il percorso formativo ha coinvolto 21 insegnanti (6 insegnanti di scuola dell'infanzia e 15 insegnanti di scuola primaria) e si è svolto da novembre 2021 ad aprile 2022. La durata è stata di 20 ore suddivise in: 4 ore di formazione iniziale sulle metodologie di problem solving e pensiero computazionale; 6 ore di progettazione ed elaborazione delle attività; 6 ore di sperimentazione delle attività

in aula con gli allievi; 4 ore di restituzione finale delle attività svolte. Le attività alla fine del corso sono state condivise tra gli insegnanti come buone pratiche per lo sviluppo da parte degli alunni di competenze di problem solving e di pensiero computazionale. L'intero corso si è tenuto online, all'interno di un ambiente digitale di apprendimento, condiviso tra docenti e formatori, creato ad hoc per svolgere le attività sincrone e asincrone del corso. Le metodologie didattiche adottate sono state spiegazioni frontali con discussioni, analisi di attività e casi studio e attività di gruppo. Nelle prime 4 ore sono state presentate le metodologie di problem solving e di pensiero computazionale, facendo riferimento alla letteratura scientifica e alle Indicazioni Nazionali. Sono stati mostrati e discussi nel dettaglio diversi esempi di attività per spiegare agli insegnanti il tipo di attività che dovevano progettare; tali esempi sono stati messi a loro disposizione in modo che potessero utilizzarli in classe con i loro alunni. Le successive 6 ore del corso sono state dedicate alla progettazione delle attività di problem solving e di pensiero computazionale. Ogni docente doveva progettare una propria attività e consegnare la relativa scheda. Per la progettazione di ogni attività gli insegnanti dovevano seguire un'apposita scheda e potevano lavorare in piccoli gruppi per collaborare con i colleghi, scambiarsi idee e avere un supporto più mirato da parte dei formatori. La scheda di progettazione oltre a chiedere agli insegnanti di indicare le caratteristiche principali dell'attività (titolo, età dei bambini, obiettivo, prerequisiti e ambientazione) proponeva di riflettere sulle difficoltà che i bambini potevano incontrare nella comprensione delle richieste e delle istruzioni impartite o nella scelta della strategia risolutiva. Nel caso delle attività di pensiero computazionale è stato anche chiesto di indicare il linguaggio in cui gli allievi dovevano esprimere le istruzioni e i principali processi mentali coinvolti nell'attività. Dopo aver sperimentato in classe le attività progettate, gli insegnanti hanno compilato un questionario per descrivere come si è svolta l'attività, le difficoltà riscontrate e l'impatto che ha avuto sui bambini, andando a delineare eventuali modifiche nel caso volessero riproporre l'attività in classe in futuro. Le ultime 4 ore del corso sono state dedicate alla restituzione e all'analisi delle sperimentazioni svolte, nell'ottica di condividere attività e buone pratiche. Agli insegnanti è stato chiesto di riflettere sulle pratiche didattiche utilizzate per favorire il problem solving e il pensiero computazionale prima e dopo il corso di formazione attraverso due questionari, uno compilato all'inizio del corso e l'altro alla fine; quest'ultimo prevedeva anche domande di gradimento del percorso formativo. Riassumendo, agli insegnanti al termine del percorso è stato chiesto di consegnare: un questionario iniziale, un questionario di gradimento finale, due schede di progettazione e un questionario relativo alla sperimentazione in classe. Per rispondere alla domanda di ricerca sono state analizzate le attività sperimentate dagli insegnanti nelle loro classi, le relative schede di progettazione e le risposte degli insegnanti ai vari questionari.

4 Metodologia

Per capire come gli insegnanti hanno lavorato per stimolare i bambini e gli alunni a sviluppare il pensiero computazionale, sono state analizzate sul totale di 21 attività le 7 attività di pensiero computazionale sperimentate in classe e i questionari relativi alla loro sperimentazione. Di queste 7 attività, 2 sono state sperimentate da docenti della scuola dell'infanzia e 5 da docenti della scuola primaria. Le attività hanno coinvolto un totale di 86 bambini, dei quali 34 della scuola dell'infanzia e 52 della scuola primaria. La riflessione dei formatori sul materiale prodotto è avvenuta a partire dal questionario compilato dagli insegnanti in seguito alla sperimentazione in classe. In particolare, sono stati presi in considerazione i seguenti aspetti del questionario:

- i processi mentali stimolati durante l'attività;
- le difficoltà rilevate dai bambini e il feedback rilasciato dall'insegnante in relazione a tali difficoltà. Oltre ad analizzare i processi mentali attivati durante le attività indicati dai docenti, ogni autrice ha indicato in modo indipendente quali potessero essere i processi mentali coinvolti. Dopodiché vi è stata una discussione di gruppo per trovare un'unica codifica. Per capire meglio le scelte effettuate dagli insegnanti è stata analizzata anche la scheda di progettazione, al fine di evidenziare eventuali analogie e differenze tra quello che essi si aspettavano e quello che è realmente accaduto. Per comprendere il grado di apprezzamento e utilità del corso è stato considerato il questionario finale compilato al termine del percorso di formazione. In riferimento al quadro teorico presentato sul pensiero computazionale, saranno mostrate le attività progettate e sperimentate in classe, andando a riflettere sulle strategie che possono essere efficaci.

5 Risultati

Le 7 attività analizzate, di cui 2 realizzate da insegnanti della scuola dell'infanzia e 5 della scuola primaria, sono state create con l'intento di aiutare i bambini, fin da piccoli, a stimolare le competenze di pensiero computazionale e problem solving. L'analisi dei questionari delle attività sperimentate mostra che tutti gli insegnanti sono concordi nel porre particolare attenzione al contesto, che deve essere il più possibile accattivante e vicino all'esperienza dei bambini, utilizzando anche situazioni ludiche, che rendano le attività più attraenti. Tutti gli insegnanti sono stati attenti a identificare nella fase di progettazione le possibili difficoltà che i bambini avrebbero potuto incontrare e hanno individuato azioni apposite per far fronte a tali difficoltà. Nella maggior parte dei casi le difficoltà immaginate in fase di progettazione si sono verificate durante la sperimentazione e sono state attenuate grazie agli strumenti che gli insegnanti avevano previsto per contrastarle. Ad esempio, una delle difficoltà maggiormente riscontrate riguarda la capacità di distinguere la destra dalla sinistra. Per aiutare gli alunni a superare tale ostacolo, alcuni insegnanti hanno legato un braccialetto di colore giallo al polso destro degli alunni e uno di colore rosso al polso sinistro.

Per quanto riguarda gli elaborati della scuola dell'infanzia, entrambi gli insegnanti hanno scelto di iniziare dal racconto di una storia, la nota favola di "Cappuccetto Rosso" in un caso e quella de "Il lupo, la pecora e il cavolo" nell'altro, che narra la storia di un pastore che, dopo aver raggiunto la riva del fiume insieme ai tre personaggi, lascia il lupo nel bosco, la pecora nel recinto della sua fattoria e infine porta a casa il cavolo, ma il mattino seguente si accorge che la pecora è scappata. Dunque, viene chiesto ai bambini di aiutare Cappuccetto Rosso in un caso e il pastore nell'altro, a ricongiungersi rispettivamente con la nonna e la pecora smarrita, cercando di schivare il lupo e i vari ostacoli presenti nel bosco. In entrambi i casi l'insegnante ha disegnato un reticolo sul pavimento, dove i bambini si sono potuti muovere e hanno potuto individuare un percorso che i due protagonisti devono compiere. In entrambe le attività, gli insegnanti hanno inserito alcuni ostacoli nel reticolo (ad esempio un albero, il lupo, ecc.), aumentandone la difficoltà e chiedendo ai bambini di tenerli in considerazione ed evitarli nel percorso da loro scelto, che non era l'unico possibile. Successivamente l'attività entra nella vera e propria fase di pensiero computazionale: i bambini a turno si spostano nuovamente sul reticolo, ma le indicazioni per ogni mossa vengono fornite da un compagno in un linguaggio specifico. Nel primo caso si utilizzano le "Codyfeet" (<https://codemooc.org/codyfeet/>), quattro tessere quadrate di colori diversi raffiguranti due piedi disposti nella direzione in cui ci si intende muovere, mentre nel secondo si utilizzano frecce colorate e numeri, che indicano rispettivamente la direzione (avanti, dietro, destra e sinistra) e il numero di volte in cui muoversi in tale direzione (Figura 1).



Figura 1

Sperimentazione dell'attività "Cappuccetto rosso" a destra, "Il lupo, la pecora e il cavolo" a sinistra

Nelle due attività vengono attivati il pensiero algoritmico e logico per individuare la sequenza corretta e ordinata di passaggi, la decomposizione, in quanto il percorso finale è costituito da una serie di passaggi che uniti compongono la sequenza finale, e l'astrazione, che consente ai bambini di focalizzarsi sull'obiettivo finale, piuttosto che su elementi di contorno come gli ostacoli

presenti sul reticolo. È stato possibile attivare il processo di generalizzazione, in quanto è stato chiesto ai bambini di riportare il percorso dal reticolo sul pavimento al reticolo disegnato dall'insegnante sul foglio di carta. Inoltre, per la prima attività, l'insegnante ha introdotto un secondo linguaggio e ha chiesto ai bambini di tradurre le indicazioni espresse in "Codyfeet" in frecce disposte nelle 4 direzioni (destra, sinistra, avanti e dietro) rispettando il colore della "Codyfeet" corrispondente (esempio in Figura 2).



Figura 2

Trascrizione del percorso nel linguaggio opportuno da parte degli alunni

Per la scuola primaria tutte le attività hanno previsto l'utilizzo di un reticolo, anche se non sempre sul pavimento. In particolare, tre attività ("Il tesoro smarrito di Capitan Mezzabarba", "L'isola dei pirati" e "Divertiamoci con il coding") richiedevano la progettazione e lo sviluppo di un percorso, così come le attività della scuola dell'infanzia, per la realizzazione di un obiettivo finale, per esempio raggiungere il tesoro. Anche in questi casi le istruzioni fornite dai compagni venivano espresse in un linguaggio specifico e gli alunni in prima persona o i compagni dovevano comprenderle, interpretarle ed eseguirle in modo tale da attivare il pensiero algoritmico, il pensiero logico, la decomposizione e l'astrazione (esempio in Figura 3).

Nelle attività "L'isola dei pirati" e "Il tesoro smarrito di Capitan Mezzabarba" è stato possibile anche attivare il processo di generalizzazione, in quanto l'insegnante ha chiesto ai bambini di riportare il percorso fatto sul pavimento su un foglio di carta (Figura 4).

Una variante particolarmente interessante di queste attività è stata l'attività "Il duello", nella quale viene richiesto di attivare anche competenze di problem solving in quanto i due giocatori che muovono una pedina sulla scacchiera devono cercare di catturarsi l'un l'altro. In questa attività, ogni giocatore ha 5 carte a disposizione, che pesca una alla volta da un mazzo in cui sono presenti 4 tipi di carte: "Gira a destra", "Gira a sinistra", "Avanti", "Indietro". Il giocatore, quando è il proprio turno, valuta le carte a sua disposizione e ne usa una o anche di più per muovere la propria pedina sulla scacchiera con l'obiettivo di catturare il compagno. Anche in questa attività, come in quelle precedenti, vengono attivati il pensiero algoritmico e logico e la decomposizione per individuare la giusta sequenza ordinata di passaggi per raggiungere l'obiettivo, e l'astrazione che consente al bambino di focalizzarsi sull'obiettivo finale, piuttosto che su elementi di disturbo presenti sul reticolo. Un'attività che si distingue dalle precedenti è l'attività "Coding con la pixel art", in cui veniva chiesto agli alunni completare delle schede fornite dall'insegnante. In un primo momento venivano fornite ai bambini delle istruzioni per colorare la loro griglia vuota correttamente; in un secondo momento veniva chiesto ai bambini di fare un disegno libero sulla griglia e quindi di scrivere le istruzioni corrette da fornire ai compagni per realizzare lo stesso disegno (Figura 5).

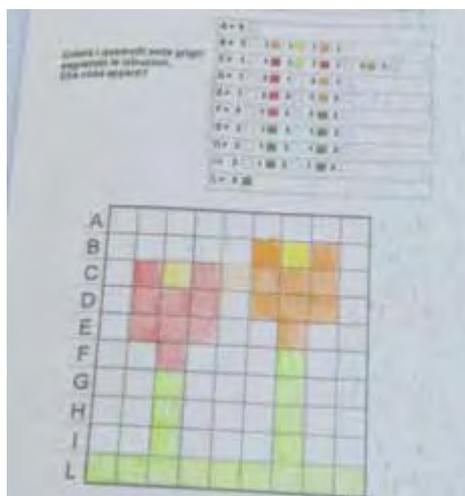


Figura 5
Scheda completata dell'attività "Coding con la pixel art"

In questa attività, oltre ad essere attivati il pensiero algoritmico, il pensiero logico, la decomposizione e l'astrazione, tramite la seconda richiesta viene stimolata anche la generalizzazione poiché gli alunni devono adattare quanto fatto nella prima richiesta per risolvere un problema nuovo. In Tabella 1 sono riportate le attività presentate e i processi mentali che vengono attivati durante un'attività di pensiero computazionale. Sono segnati con una "X" i processi mentali che gli insegnanti ritengono che l'attività stimoli, le "X" in grassetto si riferiscono ai

processi da loro non indicati, ma che potrebbero essere attivati. Dalla tabella si nota che i processi mentali maggiormente stimolati sono il pensiero algoritmico, il pensiero logico e la decomposizione, l'astrazione e la generalizzazione. La vera difficoltà si è riscontrata nel riconoscere pattern: osserviamo infatti che nessuna attività ha previsto l'attivazione di tale processo mentale. Questo potrebbe derivare dal fatto che le 7 attività non prevedevano un vero e proprio utilizzo di regolarità nei dati e nei problemi, ma nella maggior parte dei casi hanno a che vedere con lo sviluppo di percorsi espressi in linguaggi specifici, motivo per cui i primi quattro processi sono più frequenti. Probabilmente è un processo più elevato e più difficile da proporre a bambini di questa età a cui servono attività introduttive di pensiero computazionale. Potrebbe essere incluso in attività più complesse dedicate ad alunni più grandi. Dall'analisi dei questionari delle attività sperimentate in classe è emersa la difficoltà riscontrata dagli insegnanti nel comprendere a fondo il significato dei processi mentali e la loro differenza. Infatti, come si può osservare in Tabella 1 nessun insegnante ha individuato il processo di generalizzazione, probabilmente per una difficoltà nel capire di cosa si tratta.

Titolo attività	Processi mentali					
	Pensiero algoritmico	Pensiero logico	Decomposizione	Astrazione	Riconoscere pattern	Generalizzazione
Il lupo, la pecora ed il cavolo	x	x	x	x		x
Il percorso Rosso	x	x	x	x		x
Il tesoro smarrito di Capitan Mezzabarba	x	x	x	x		x
L'isola dei pirati	x	x	x	x		x
Divertiamoci con il coding	x	x	x	x		
Il duello	x	x	x	x		
Coding con la pixel art	x	x	x	x		x

Tabella 1

Descrizione dei processi mentali attivati durante le attività

Nel questionario finale ai docenti è stato chiesto di valutare diversi aspetti del percorso formativo, su una scala da 1 (per niente) a 5 (molto). In particolare, i

docenti hanno apprezzato: la chiarezza delle spiegazioni (in media 4.4), la spendibilità in classe delle metodologie proposte (4.4), il supporto dei formatori (4.5), l'interazione con i formatori (4.5). Frase gradimento corso e gradimento attività con i bambini. Alla domanda aperta "Quali metodologie proposte in questo corso hai già utilizzato o intendi utilizzare con le tue classi?" alcune risposte ricorrenti sono state: "Ho usato un'attività ideata da un collega, adattandola alla mia classe", "Lavorerò sicuramente di più sul pensiero computazionale", "Ho svolto in classe con soddisfazione una delle attività presentate dai formatori". Dai pareri emersi dagli insegnanti durante l'incontro di restituzione finale è emerso che i bambini e gli alunni coinvolti hanno apprezzato molto le attività proposte dai loro insegnanti, si sono divertiti e allo stesso tempo hanno appreso nuovi contenuti e stimolato le loro capacità di pensiero computazionale.

6 Conclusioni

In questo articolo sono state analizzate le attività di pensiero computazionale progettate e sperimentate dagli insegnanti durante il "Percorso di formazione per gli insegnanti: progettare e valutare attività per lo sviluppo di competenze di problem solving e pensiero computazionale". Per rispondere alle domande di ricerca, abbiamo analizzato le attività progettate dagli insegnanti, le schede di progettazione e le risposte ai vari questionari. Le tipologie di attività presentate hanno permesso di sviluppare competenze di pensiero computazionale e di attivare soprattutto i processi mentali di pensiero algoritmico, pensiero logico, decomposizione e astrazione. Le scelte degli insegnanti sono state effettuate nell'ottica di realizzare esempi che potessero essere adatti ai loro allievi e che allo stesso tempo consentissero lo sviluppo di competenze di pensiero computazionale e di problem solving. Ad esempio, la totalità degli insegnanti ha preferito un'ambientazione e un contesto accattivante per i bambini e in alcuni casi si è scelto di utilizzare il gioco come strumento di apprendimento. L'impatto in classe è stato positivo e i bambini coinvolti hanno apprezzato molto le attività ideate dai loro insegnanti, si sono divertiti e allo stesso tempo hanno appreso nuovi contenuti e stimolato le loro competenze di pensiero computazionale. Dal questionario finale è emerso che gli insegnanti hanno trovato utile il corso di formazione, in particolare le attività e le metodologie proposte dai formatori. Grazie al corso di formazione, gli insegnanti hanno affermato di aver acquisito maggiore consapevolezza delle strategie didattiche sul pensiero computazionale e sul problem solving e di aver appreso nuove competenze nella progettazione delle attività. Vista la difficoltà degli insegnanti a comprendere in maniera approfondita il significato dei processi mentali attivati nel pensiero computazionale e la loro differenza, in un futuro corso di formazione si intende prestare maggiore attenzione a questo aspetto. Gli insegnanti hanno apprezzato particolarmente le attività dei loro colleghi e ne hanno sperimentate alcune nella loro classe, oppure hanno considerato di utilizzarle in futuro. Per il futuro, si potrebbe pensare di creare un database di attività sul problem solving e sul

pensiero computazionale che possa essere condiviso in un'ampia comunità di insegnanti, in una prospettiva verticale, come la comunità del PP&S.

Bibliografia

Barana, A., Conte, A., Fissore, C., Floris, F., Marchisio, M., & Sacchet, M. (2019). The creation of animated graphs to develop computational thinking and support STEM education. *In Maple Conference* (pp. 189-204). Springer, Cham.

Barana, A., Fissore, C., Marchisio, M., & Pulvirenti, M. (2020). Teacher training for the development of computational thinking and problem posing & solving skills with technologies. *In The International Scientific Conference eLearning and Software for Education* (pp. 136-144), 2. Carol I National Defence University Publishing House.

Città, G. et al. (2021). Pensiero Computazionale ed Embodied Cognition. *In Atti Convegno Nazionale DIDAMATiCA 2021* (pp. 185-191). AICA-Associazione Italiana per l'Informatica ed il Calcolo Automatico.

European Commission. (2016). *The European Digital Competence Frameworks for Citizens*. Luxembourg: Publications Office of the European Union.

Fissore, C., & Marchisio, M. (2020). Didattica Online a distanza durante l'emergenza da Covid-19: un Ambiente Digitale di Apprendimento per le Digital Humanities. *In Didamatica2020* (pp. 346-355). AICA-Associazione Italiana per l'Informatica ed il Calcolo Automatico.

Kalelioğlu, F., Gülbahar, Y., & Kukul, V. (2016). A framework for computational thinking based on a systematic research review. *In Baltic J. Modern Computing* (pp. 583-596), 4.

Lodi, M., Martini, S., & Nardelli, E. (2017). Abbiamo davvero bisogno del pensiero computazionale? *In Mondo Digitale* (pp. 1-15), 72.

M Lodi, M., Martini, S., Sbaraglia, M., & Zingaro, S. P. (2020). (Non) parliamo di pensiero computazionale. Negli atti del XXXV Convegno Nazionale "Incontri con la Matematica", Castel San Pietro Terme/online, 5-6-7 novembre 2021.

MIUR. (2012). *National Guidelines for the kindergarten school curriculum and the first cycle of education*.

Papert, S. (1980). *Mindstorms: Children, Computers and Powerful Ideas*. New York: Basic Books, Inc.

Wing, J. M. (2006). Computational thinking. *In Communications of the ACM* (pp. 33-35), 49(3).

Wing, J. M. (2017). Computational thinking's influence on research and education for all. *In Italian Journal of Educational Technology* (pp. 33-35), 25 (2).

Futuri insegnanti di sostegno. Gli Smart Visual Media per l'inclusione scolastica degli alunni con disabilità

Pierluigi Muoio¹

¹ Università della Calabria, Via P. Bucci, 87036 Arcavacata di Rende (CS)
pierluigi.muio@unical.it

Sommario

L'articolo vuole proporre uno spunto di riflessione sulle potenzialità didattiche delle tecnologie emergenti di tipo immersivo quali ambienti di apprendimento a 360 gradi, dinamici, accattivanti ed efficaci tali da consentire nuove esperienze di insegnamento- apprendimento. In particolare ci si sofferma su uno Smart Visual Media progettato e realizzato nell'ambito dei corsi di formazione per il conseguimento della specializzazione per il sostegno didattico presso l'Università della Calabria. Il prodotto multimediale è stato successivamente erogato in una classe quinta di scuola primaria con l'obiettivo di promuovere l'inclusione e stimolare l'apprendimento. La scuola di oggi, investita da nuove responsabilità e dalle diverse sollecitazioni provenienti dall'esterno, è obbligata ad essere accessibile non solo dal punto di vista strutturale ed ambientale, ma anche rispetto ai contenuti presentati in classe. I nuovi strumenti devono agevolare altrettanti nuovi modelli e metodologie di insegnamento per assicurare ad ogni allievo la partecipazione al processo di insegnamento-apprendimento, senza escludere nessuno. Insegnanti competenti, formati ed aggiornati sulle nuove opportunità, anche di tipo tecnologico, sono indispensabili per rendere la scuola una casa della cultura capace di accogliere con competenza le singolarità consentendo la pari partecipazione agli allievi svantaggiati. Nell'ottica di un'inclusione possibile e doverosa una risorsa dal valore fondamentale è rappresentata dall'insegnante di sostegno, figura chiave nel rapporto docente-allievo e scuola-società.

1 Introduzione

Il contributo descrive la progettazione e la realizzazione di uno Smart Visual Media, ovvero un oggetto didattico multimediale e inclusivo basato sulle immagini e sulla realtà aumentata, realizzato al termine del percorso di specializzazione per il sostegno didattico presso l'Università della Calabria. Il Corso, previsto dal D.M. n. 249/2010, disciplinato dal D.M. 30 settembre 2011, prevede nella sua organizzazione lo svolgimento di 75 ore dedicate alle TIC, pari a complessivi 3 Crediti Formativi Universitari (CFU), rientranti nello spazio dedicato al tirocinio indiretto. In sede di esame finale i candidati devono dimostrare l'acquisizione delle competenze didattiche legate all'uso delle tecnologie presentando un prodotto multimediale correlato all'esperienza diretta di tirocinio, in grado di facilitare l'acquisizione dei contenuti da parte degli allievi con bisogni educativi speciali, stimolare la partecipazione attiva di tutti gli alunni e favorire contesti di apprendimento pienamente inclusivi. La figura dell'insegnante di sostegno è una professionalità articolata il cui iter formativo ha registrato sostanziali cambiamenti negli ultimi anni. Si tratta di un professionista della complessità (Montanari & Ruzzante, 2020), garante dei diritti di tutti e di ciascuno, in particolare degli allievi più fragili, e figura chiave nell'ambito del processo di apertura del sistema scolastico ai soggetti con disabilità. La presenza nelle classi e nelle scuole ordinarie richiede all'insegnante di sostegno una formazione più approfondita e precisa nell'esperienza pratica per poter rispondere alle diverse esigenze degli alunni, considerato che la formazione iniziale risulta essere troppo generale e orizzontale per poter essere all'altezza delle diverse situazioni. Nel dinamico contesto della società odierna la scuola è chiamata al non facile compito di rimuovere le barriere che ostacolano l'apprendimento e la partecipazione di tutti gli allievi alla vita scolastica, azzerando le differenze linguistiche, culturali, religiose, geografiche esistenti. In tale quadro gli insegnanti, in particolar modo quelli di sostegno, assumono un ruolo determinante, in quanto agenti strategici dei processi di inclusione scolastica e sociale, e a loro viene richiesto un costante investimento formativo per poi essere in grado di innovare e differenziare le proposte didattiche affinché possano emergere e maturare le potenzialità di ognuno. La scuola, insieme alle diverse agenzie educative presenti sul territorio, tra i suoi compiti ha anche quello di favorire l'attiva e piena partecipazione ai processi di apprendimento dei soggetti con bisogni speciali, abbattendo ostacoli, barriere, pregiudizi, e sostenere la condivisione all'interno del setting educativo di esperienze ed apprendimenti con i propri pari. La Dichiarazione di Salamanca (UNESCO, 1994), manifesto della scuola inclusiva, proclama che "le persone con bisogni educativi speciali devono poter accedere alle normali scuole, le quali devono integrarli in un sistema pedagogico centrato sul bambino" e invita i governi ad "adottare, come legge o politica, il principio dell'educazione inclusiva, accogliendo tutti i bambini nelle scuole normali". Le diverse categorie di bisogni speciali, generati da una vasta gamma di condizioni e fattori differenti (lanes, 2005), possono essere manifestati da tutti gli allievi a seguito di

difficoltà temporanee o permanenti, pertanto nella scuola inclusiva la classe rappresenta un insieme di allievi per sua natura eterogeneo e le differenze diventano l'ordinaria caratteristica nello svolgimento delle pratiche quotidiane in aula (Pavone, 2004). Le complesse problematiche proposte dalla società liquida contemporanea (Bauman, 2020) impongono di rivedere la classica distinzione tra allievi "speciali" e allievi "normali", con il conseguente pensiero di ricorrere a soluzioni e risorse straordinarie per i primi e a soluzioni e risorse ordinarie per i secondi. La realtà, nella quale gli allievi speciali hanno anche bisogni normali e gli allievi normali possono richiedere bisogni speciali, richiede la predisposizione di contesti in grado di raggiungere tutti gli studenti e offrire medesime occasioni di crescita per chi vive particolari situazioni di difficoltà. Si tratta di obiettivi e valori generalmente accettati a livello internazionale, come indicato dal Rapporto sulla Disabilità dell'Organizzazione Mondiale della Sanità (OMS, 2011) e le Conclusioni del Consiglio dell'Unione Europea sulla Dimensione Sociale dell'Istruzione e della Formazione (2010), che devono condurre a sistemi scolastici in grado di garantire equità e merito senza creare categorizzazioni e discriminazioni. Si tratta di far viaggiare parallelamente, creando una sinergia virtuosa, da una parte la direttrice delle politiche inclusive e dall'altra quella della formazione dell'insegnante inclusivo, ed in particolare quello di sostegno, per rispondere alle carenze di condivisione e collaborazione presenti nella realtà scolastica (D'Alonzo, 2012) e costruire prospettive sempre migliori di vita e benessere personale e sociale, al fine di creare una società inclusiva in cui assicurare a tutti le pari opportunità.

2 Insegnante di sostegno e formazione

La formazione costituisce un elemento imprescindibile per il docente moderno chiamato ad agire all'interno di una società che impone all'individuo di doversi misurare con regole e situazioni sempre nuove e cangianti, di abbandonare le vecchie e consolidate certezze che lo sostenevano nel vivere quotidiano del passato, facendo diventare il sapere un bisogno fondamentale dell'uomo. La formazione, iniziale e permanente, risulta da un lato funzionale alla piena realizzazione e allo sviluppo della professionalità, e dall'altro rappresenta una delle fondamentali dimensioni della professione docente. Insegnanti competenti, formati ed aggiornati sulle nuove opportunità, anche di tipo tecnologico, sono indispensabili per rendere la scuola una casa della cultura capace di accogliere con competenza le singolarità, nell'ottica di un'inclusione possibile e doverosa. Oggi più che mai, il concetto di formazione non coincide più con un determinato gruppo di contenuti, per di più acquisiti nella fase dell'infanzia e della giovinezza, ma è sempre più legato ai processi dinamici dei soggetti, ed è riferito alle dimensioni culturali dello sperimentare, del comprendere e del valutare, unite alla capacità di prendere ed assumere decisioni in modo consapevole ed autonomo (Piu, 2007). La formazione nel caso dell'insegnante di sostegno assume ancora maggiore

importanza, essendo esso figura strategica con ruolo pivotale (Cottini, 2014) in grado di percepire bisogni ed emergenze educative, di ridisegnare i percorsi rispettosi di differenze e diversità, contribuendo all'innovazione di contesti e pratiche. L'iter formativo previsto per l'insegnante di sostegno è stato oggetto di diversi mutamenti negli ultimi decenni. Istituita con la Legge 517 del 1977, tale figura assume la contitolarità delle classi in cui opera, partecipa alla programmazione educativa e didattica e alla elaborazione e verifica delle attività di competenza dei consigli di interclasse, dei consigli di classe e dei collegi dei docenti. Spesso considerato erroneamente insegnante di secondo piano, l'insegnante specializzato è titolare di diritti e di doveri allo stesso modo di quello curricolare, e con pari dignità professionale svolge compiti i cui effetti hanno ricaduta sia nella sfera del rapporto docente-alunno, sia in quella più ampia del rapporto scuola-società. È regista e promotore di buone prassi con l'obiettivo di accompagnare l'allievo con disabilità nel percorso formativo e in tale ottica coinvolge tutto il team docente (di cui fa parte) nel pensare, progettare e valutare itinerari didattici in grado di ridurre le condizioni di svantaggio e valorizzare le potenzialità di ogni alunno. La positiva interazione tra il docente di sostegno e docenti curricolari è dunque condizione imprescindibile per il successo del processo di inclusione, pertanto va rifiutata l'idea di una relazione didattica individuale e separata insegnante di sostegno-allievo disabile, collegata ad una visione assistenziale e pietistica di tale figura che conduce ad un aumento dei fenomeni di delega ed isolamento. L'insegnante specializzato, pertanto, deve saper diffondere pratiche inclusive dopo aver interpretato con sensibilità e tatto i bisogni educativi del gruppo classe e le differenti abilità degli alunni, coltivando fertili rapporti di collaborazione con i colleghi su posto comune al fine di progettare efficaci azioni di insegnamento- apprendimento. La ricchezza di compiti, azioni e responsabilità fin qui esaminati sollecita molteplici dimensioni della professionalità docente e richiede l'acquisizione, lo sviluppo ed il consolidamento di abilità e competenze polivalenti attraverso una formazione che permetta di acquisire una varietà di strategie e metodologie didattiche e approcci culturali. Il percorso di specializzazione universitario disegnato dal D.M. n. 249/2010 ha la durata di un anno accademico, pari a 60 CFU, ed è strutturato in insegnamenti, laboratori, tirocinio (diretto ed indiretto) e prevede una prova di accesso e una finale. Tirocini e laboratori sono articolati diversamente in base al grado scolastico (infanzia, primaria, secondaria di primo grado, secondaria di secondo grado). Il Corso si propone di fornire una formazione sistematica e intenzionale, trasferendo competenze educative-didattico adeguate a garantire un lavoro qualificato ed innalzare il gradiente inclusivo della scuola intesa come autentica comunità educante (Gaspari, 2019). L'accento è posto oltre che sulle competenze pedagogiche anche sulla dimensione relazionale da spendere in contesti (macro-micro-meso) integrati anche fuori dalla classe (Bochicchio, 2017) al fine di accreditare l'insegnante specializzato quale alter ego maior dell'insegnante speciale del passato (Zappaterra, 2014). A

supporto della propria azione, il docente di sostegno si avvale di aiuti e strumenti tra i quali quelli tecnologici che offrono adeguate opportunità e molteplici occasioni per favorire l'apprendimento e raggiungere obiettivi di inclusione, partecipazione, autonomia ed accessibilità economica al sapere digitale.

3 La Realtà Aumentata nella didattica

La Realtà Aumentata (Augmented Reality, AR) è un insieme di tecnologie che migliorano la percezione della realtà, sovrapponendo ad essa oggetti o informazioni virtuali sensibili al contesto (Klopfer & Sheldon, 2010). Si colloca lungo il continuum tra il "completamente reale" e il "completamente virtuale" (Milgram & Kishino, 1994) e negli ultimi venti anni si è diffusa nei diversi ambiti della vita comune, rendendosi disponibile sia tramite PC che su dispositivi mobili, nonché con dispositivi dedicati come visori, caschi e occhiali. Nel contesto didattico, l'utilizzo della realtà aumentata rappresenta una preziosa opportunità per innovare i processi di insegnamento/apprendimento, per coinvolgere e promuovere negli allievi l'acquisizione delle competenze digitali consentendo loro di applicare i concetti teorici in contesti reali o verosimili. La realtà aumentata consente di generare ambienti di apprendimento digitali di nuova generazione che si fondono con i luoghi fisici tradizionalmente deputati alla formazione, aumentando le possibilità di condivisione, interazione e socializzazione. In tal modo gli allievi possono immergersi e fare esperienze con un alto grado di coinvolgimento socio emotivo. La tecnologia AR ha la capacità di rendere in modello 3D tutto ciò che può essere difficilmente visualizzato in una classe, al computer e nella mente degli studenti. I contenuti astratti e difficili da immaginare diventano visibili e interattivi, il che favorisce una migliore comprensione degli argomenti della lezione (Sural, 2018). Uno dei principali punti deboli del sistema scolastico a tutti i livelli, la mancanza di realismo (Kearsley, 2000), può essere superato da una didattica che offre esperienze maggiormente autentiche grazie ad una più stretta connessione tra la conoscenza trasmessa dal docente e le pratiche della vita reale. L'ampia diffusione di applicazioni e piattaforme in grado di manipolare e generare contenuti multimediali, comprese quelle di realtà aumentata, insieme all'aumento della familiarità nell'uso di strumenti 2.0, in gran parte disponibili sul Web senza la necessità di installazione in "locale", ha dato una spinta propulsiva alla produzione autoriale tra gli insegnanti. La Realtà Aumentata consente loro di esplorare approcci metodologici diversi, in modo da superare la trasmissione di semplici nozioni e offrire un apprendimento interattivo attento a bisogni e aspettative degli allievi. È compito dell'insegnante, infatti, predisporre percorsi didattici e produrre materiali tenendo in considerazione che nel contesto in cui opera si manifestano necessità ed esigenze diverse tra loro, accogliendo e concedendo a tutti le stesse opportunità. La prospettiva inclusiva è basata sul riconoscimento delle differenze di ciascuno e sulla necessità che a ciascuna differenza venga

riconosciuta piena legittimità. Sotto questo aspetto la realtà aumentata ha elevate potenzialità di inclusione considerato che stimola e sollecita simultaneamente più sensi all'interno dell'esperienza di apprendimento, venendo incontro alle esigenze dei diversi stili di apprendimento e ai bisogni educativi degli allievi con disabilità (Di Martino & Longo, 2019). Questa tecnologia, considerata come ambiente di apprendimento ibrido e immersivo (Akçayir & Akçayir, 2017) promuove nei discenti la motivazione, l'acquisizione di abilità di ricerca, l'esplorazione del mondo reale, lo sviluppo del pensiero critico e delle capacità di problem solving. Inoltre, la AR promuove e supporta l'apprendimento collaborativo e favorisce l'interazione sociale tra studenti permettendo loro di essere co-artefici del proprio percorso di formazione, come evidenziato da esperienze di co-progettazione di libri di testo aumentati (Alhumaidan, Pui Ying Lo, & Selby, 2018). Il tutto tenendo presente dell'importanza dell'accessibilità all'informazione e alla comunicazione, ribadita anche nella Convenzione ONU sui diritti delle persone con disabilità approvata il 13 dicembre 2006 che indica tra i diritti riconosciuti loro anche l'accessibilità delle tecnologie informatiche. In tale prospettiva il docente è tenuto a considerare usabilità ed accessibilità quali fattori discriminanti nel selezionare, di volta in volta, gli strumenti tecnologici e i materiali didattici, anche autoprodotti, più adeguati alle diverse situazioni e quindi compatibili con le necessità degli allievi, da inserire in progettazioni inclusive che abbiano sullo sfondo la logica della partecipazione, nello spirito della Progettazione Universale (Mace, 1985) e dello Universal Design for Learning (Rose & Meyer, 2002). Le tecnologie ricoprono il ruolo di strumenti compensativi, permettendo al soggetto svantaggiato di svolgere azioni che altrimenti sarebbero precluse oppure di compierle con minore fatica e con più padronanza attraverso la possibilità di attivare percorsi alternativi ma equivalenti. Un buon insegnante specializzato sarà in grado di integrare nelle sue scelte didattiche e progettazioni inclusive strumenti, tecnologie, applicazioni, ausili, nella misura in cui essi permettono di rimuovere gli ostacoli alla partecipazione e all'apprendimento, considerandoli come vettori per attuare una vera didattica inclusiva, interattiva e coinvolgente diretta a tutta la classe e non esclusivamente ad allievi con disabilità. L'eterogeneità all'interno della classe, la presenza di alunni diversi e il conseguente agire in un contesto educativo ricco, come nel caso descritto nel presente lavoro, invita il docente a porsi delle domande per saper cogliere le singolari modalità di funzionamento di ciascuno. Le tecnologie, pertanto, supportano l'insegnante nell'adozione e sperimentazione di modelli, prassi e strategie didattiche per valorizzare tempi e modi personali di ognuno in un clima motivante, cooperativo e creativo.

4 Il contesto scolastico e l'allievo

Il prodotto multimediale, descritto nel successivo paragrafo, è stato ideato nell'ambito delle ore di tirocinio diretto presso un Istituto Comprensivo con

sede a Roma. La classe si trova al secondo piano di una sede distaccata dell'Istituto ed è composta da 18 alunni, di cui 6 femmine e 12 maschi. Il livello degli alunni è eterogeneo, sono presenti tre alunni BES (ai sensi della D.M. del 27/12/2012) così identificati: Disturbo specifico di apprendimento certificato (L.170/2010), diagnosticato come discalculia, Disturbi evolutivi specifici diagnosticato come FIL, Funzionamento Intellettivo Limite o "borderline cognitivo" e svantaggio socio-economico. Non sono presenti stranieri ed italofofoni. La classe è dotata dello strumento multimediale LIM collegato in rete. Il tempo scuola è articolato in 40 ore con il tempo mensa giornaliero. Il tirocinio è stato svolto all'interno di una classe quinta, indirizzo primaria. L'alunno, L., ha dieci anni, presenta un Disturbo dello Spettro autistico ad alto funzionamento (certificato a norma della L. 104/1992), è verbalizzato e collaborativo. La compatibilità caratteriale fra l'allievo e la classe ha permesso di impostare un intervento educativo comune, salvaguardando differenziazioni, inclusioni ed ampliamenti di programmi individuali, quando necessari. La famiglia è composta da 4 persone: padre, madre e fratello di anni 8. Il livello socio-culturale si posiziona nella fascia medio - alta. Le ore di sostegno assegnate all'alunno sono 22, distribuite in base alle esigenze di L. e della classe. In generale gli alunni presentano disponibilità all'apprendimento e il clima in classe è sereno e costruttivo. Il processo di inclusione all'interno del gruppo classe è ben avviato, tuttavia L. preferisce relazionarsi solo con pochi compagni. Nei confronti degli insegnanti l'alunno manifesta le sue preferenze ma non ha particolari problemi a svolgere le attività con un compagno piuttosto che con l'altro, ciò grazie anche alla linea comune d'intervento che viene adottata da parte di tutti i docenti del team. L'alunno frequenta con regolarità e partecipa volentieri a tutte le attività previste per il gruppo classe, anche se mostra predilezione per talune attività; presenta un linguaggio costituito da parole-frasi che non sempre utilizza in maniera corretta. Nonostante emerga la capacità di elaborare un resoconto ragionevole di un evento di routine ha bisogno di varie domande specifiche per una rielaborazione esaustiva. Mostra responsività nella maggior parte delle interazioni sociali, tuttavia, le aperture sociali risultano, comunque, essere qualitativamente inusuali e risultano limitate a interessi personali. L'alunno mostra difficoltà a mantenere i turni di conversazione; presenta comportamenti stereotipati; necessita della guida dell'adulto per mantenere livelli di attenzione e concentrazione congrui e ciò influisce sulla qualità delle performances e sui tempi di comprensione, analisi e sintesi; non è completamente autonomo nel gestire i processi cognitivi necessari per ottenere apprendimenti significativi. Ben avviata risulta l'autonomia personale. Invece, in riferimento all'autonomia scolastica, L. necessita di una guida nell'organizzazione del lavoro. Nell'ambiente scolastico e negli spazi noti si muove con disinvoltura.

L. segue le attività didattiche della classe e riesce a portarle a termine con il supporto del docente. Il team classe ha convenuto intervallare momenti di attività in classe a momenti di lavoro, in piccolo gruppo o individualizzato,

fuori o dentro la stessa con il supporto di due, tre compagni (Peer tutoring). Dopo attenta osservazione della classe e dell'alunno è stata progettata un'attività che permettesse a tutta la classe di apprendere e lavorare insieme al fine di promuovere l'inclusione e soprattutto per L. che stimolasse la motivazione ad apprendere. Il prodotto realizzato al termine è uno Smart Visual Media, sotto forma di tour virtuale multimediale a 360 gradi, rivolto a tutta la classe al fine di veicolare un contenuto riguardante la storia.

5 Progettazione e realizzazione del prodotto multimediale

Durante le ore di tirocinio diretto è stato progettato, e successivamente erogato, uno Smart Visual Media incentrato sul tema "Alla scoperta dei romani". Per realizzare il prodotto la scelta è caduta su Thinglink, una Web App che consente di creare artefatti in realtà aumentata, originali e creativi, tra i quali video interattivi, immagini e mappe parlanti, libri aumentati, tour virtuali a 360 gradi in grado di agevolare l'adozione di approcci didattici innovativi e lo svolgimento di esperienze progettuali che stimolano la collaborazione in ottica socio costruttivista. Partendo da una immagine o da un video, Thinglink consente di "aumentare" l'elemento di base aggiungendo dei tag, ovvero dei punti di interesse, contenenti qualsiasi tipo di contenuto digitale in grado di offrire ulteriori risorse informative. Finalità del prodotto realizzato è quella di creare un clima classe accogliente, sereno, aperto, capace di includere ciascun alunno e i suoi bisogni speciali, compreso L., il bambino seguito durante il tirocinio diretto. Il prodotto è stato ideato con l'obiettivo di contribuire a creare intorno a lui un'atmosfera che prescindendo dalla sua disabilità lo incitasse a partecipare serenamente alle attività previste insieme ai compagni, in modo da ottenere una ricaduta significativa sulla sua autostima e la fiducia in sé stesso. Più nello specifico, gli obiettivi che hanno accompagnato la realizzazione del prodotto sono stati i seguenti:

- Sviluppare la capacità di collocare nello tempo gli eventi, individuando i possibili nessi tra eventi storici e caratteristiche geo-morfologiche di un territorio;
- Sensibilizzare gli alunni al concetto di tutela e salvaguardia delle opere d'arte e dei beni archeologici e di un contesto economico caratterizzato da numerose risorse naturali;
- Contribuire a migliorare le conoscenze sul tema oggetto d'indagine mediante l'ausilio delle tecnologie didattiche;
- Mettere a punto un percorso che rispetti e valorizzi le esigenze di ciascuno.

Facendo ricorso alla didattica immersiva, quindi, è stato creato un Tour virtuale nelle antiche rovine della città di Roma, mediante una guida audio e utilizzando foto a 360° che ha consentito agli alunni di entrare in contatto con l'opera di interesse e "passeggiare" per il percorso prescelto utilizzando la Realtà Aumentata, indispensabile per scoprire dettagli che solo in loco è possibile scorgere avvalendosi, anche, di una voce narrante allo scopo di

guidare il visitatore/alunno durante il tour. Il lavoro, pur presentando una propria unitarietà, può essere scomposto in alcune principali macro fasi di lavorazione. La prima è quella dell'individuazione della tematica e della ricerca dei contenuti. L'idea progettuale è stata quella di facilitare un argomento relativo alla disciplina storia, collegandolo alle tecnologie didattiche, in modo da proporre alla classe un'esperienza immersiva, un viaggio immaginario, nell'antica Roma per conoscere ed analizzare meglio un popolo, quello degli antichi romani, le sue origine, le sue leggende, i personaggi, le usanze ed il patrimonio artistico e architettonico. In questa prima fase la ricerca dei contenuti multimediali è stata condotta tenendo conto delle regole sul copyright. A tal proposito il reperimento delle immagini e degli elementi grafici è avvenuto ricorrendo a risorse rilasciate con licenza Creative Commons presenti in appositi repository presenti in Rete. La seconda fase ha visto la produzione di una serie di oggetti didattici tramite applicazioni differenti e successivamente inclusi all'interno dell'immagine aumentata. Si tratta di oggetti ed elementi realizzati pensando a differenti modalità di rappresentazione della conoscenza, in modo da non limitarsi al solo formato testuale. I vari temi della narrazione sono stati sviluppati sia sotto forma di testo, per poi individuare audio, immagini, filmati in modo da rivolgersi ai diversi stili cognitivi presenti tra gli alunni e creare un interesse per l'argomento. A tal proposito rientrano alcuni oggetti interattivi di natura ludica, realizzati ricorrendo ad applicativi quali Wordwall e LearningApps, in modo da introdurre dinamiche e meccaniche tipiche della gamification (Deterding, Dixon, Khaled, & Nacke, 2011) sempre più spesso sperimentate in ambito scolastico e universitario (Muio, 2020). La terza fase è quella della progettazione dell'interfaccia e degli strumenti di interazione tramite l'inserimento dei punti di interesse (tag). Fase di importanza centrale affinché il prodotto potesse risultare accessibile e inclusivo. Il contenuto è stato sviluppato in "scene" multiple, ovvero diverse immagini interattive e navigabili a 360 gradi, collegate tra di loro in modo da formare un percorso unitario ed immersivo. Le funzionalità di Thinglink hanno consentito di arricchire ogni scena con contenuti multimediali di varia natura (immagini, audio, video, quiz, mappe concettuali, brevi testi di spiegazione) accompagnati dalla voce dell'insegnante e dalla possibilità di ricorrere alla sintesi vocale, utile non solo per allievi DSA, ma anche per allievi normodotati che per qualsiasi motivo in un dato momento non volessero seguire il percorso tramite la lettura. L'ultima attività di progettazione ha riguardato la predisposizione di alcuni piccoli moduli interattivi di natura ludica in modo da permettere all'insegnante e agli alunni di valutare, ed autovalutare, le conoscenze acquisite durante l'esplorazione del prodotto. Al termine di tutte le fasi, pertanto, il risultato è stato quello di proporre uno Smart Visual Media (Figura 1), consultabile oltre che per mezzo di PC e dispositivi mobili (<https://bit.ly/muiodidamatica2022>), anche con l'utilizzo di visori e caschi VR con i quali sperimentare esperienze immersive.

**Figura 1**

La scena iniziale del prodotto realizzato

Terminate le fasi di progettazione e realizzazione il prodotto è stato proposto in presenza, a tutta la classe attraverso l'uso della Lavagna Interattiva Multimediale (LIM) facendo interagire gli alunni direttamente con l'oggetto didattico. La classe ha esplorato i contenuti didattici presenti sulla grafica di sfondo costituita da un'immagine evocativa la storia dell'Antica Roma. Ad ogni personaggio presente sono collegati dei tag, rappresentati da icone di vario tipo e colore, cliccando sui quali si ottengono informazioni e contenuti aggiuntivi che "aumentano" l'immagine rendendola accattivante e stimolante per tutti gli allievi. L'esplorazione virtuale inizia con audio introduttivo del docente che conduce il viaggio virtuale tramite un racconto narrato. Le icone numerate progressivamente guidano gli allievi alla scoperta dei contenuti, che di volta in volta prendono forme diverse, previste in fase di progettazione, e si rendono visibili con un clic sugli oggetti grafici corrispondenti. Gli argomenti oggetto di approfondimento sono stati: le origini di Roma e le sue leggende, gli accampamenti, le Domus e le Insulae, il Colosseo, il Circo Massimo, la famiglia e la scuola nell'Antica Roma, il sistema numerico, la tabula, i sette Re, Giulio Cesare, le divinità. Dalla scena iniziale, contenente i contenuti didattici elencati, gli allievi hanno potuto visitare virtualmente altre luoghi di interesse come il Colosseo, la Fontana di Trevi, i Fori imperiali (Figura 2) e la Cappella Sistina rappresentate tramite foto a 360 gradi completamente navigabili nelle diverse angolazioni.

**Figura 2**

La scena dedicata ai Fori Imperiali

Il momento conclusivo dell'esperienza ha previsto la verifica delle conoscenze e competenze degli allievi utilizzando oggetti di natura ludica realizzati con applicazioni quali Learning Apps, Wordwall e Kahoot. Durante tutto il tempo dell'erogazione è stata posta attenzione al livello di gradimento e partecipazione degli alunni effettuando delle rilevazioni costanti in merito alle eventuali difficoltà incontrate dagli stessi così da intervenire tempestivamente sul percorso didattico contestualizzandolo in merito alle problematiche. Gli allievi sono stati coinvolti in forme di comunicazione e interazione in cui verificare collettivamente l'apprendimento degli aspetti più rilevanti, ed esprimere singolarmente quanto compreso tramite riassunti, schemi, report interattivi. In questo modo sono state sollecitate l'interdipendenza positiva, la responsabilità individuale e di gruppo. In base ai feedback ricevuti ed alle osservazioni effettuate durante e dopo l'erogazione del prodotto si evince che l'attività si è trasformata in un momento di laboratorio suscitando interesse e partecipazione. Il prodotto ha incoraggiato l'apprendimento in tutti gli allievi ed ha favorito lo sviluppo dell'immaginazione e l'arricchimento del vocabolario personale, contribuendo ad allenare la soglia di attenzione e di conseguenza la personale capacità di ascolto. La particolare attenzione all'inclusione, fin dalla prima fase di progettazione, gli strumenti presenti nel software di produzione e le strategie adottate in classe hanno reso la narrazione dell'argomento accessibile e fruibile per tutti, adeguandolo alle esigenze personali di ciascuno. La funzione immersive reader presente in Thinglink ha reso ancora più inclusivo il prodotto finale, in quanto i testi sono immediatamente accessibili tramite uno strumento di lettura integrato che consente la traduzione automatica in oltre 60 lingue in modo da migliorare la comprensione per allievi stranieri o con dislessia. Inoltre la funzionalità di sintesi vocale integrata, tramite la quale selezionare il testo da leggere ad alta voce, ha consentito di affiancare il canale uditivo a quello visivo attivando una lettura bimodale. Ciò ha reso l'esplorazione virtuale più piacevole, contribuendo alla motivazione ed allo sviluppo della capacità di ascolto e del tempo di attenzione.

6 Conclusioni

Sollecitata da varie istanze e posta dinnanzi ai cambiamenti indotti dalla digitalizzazione, la scuola, intesa nel suo complesso, è chiamata a dotarsi di strumenti, competenze e risorse umane idonee per poter attuare una didattica realmente inclusiva, promuovendo un rinnovamento dal punto di vista metodologico, modificando tempi e spazi di apprendimento e mettendo a disposizione di tutti le tecnologie più opportune a supporto della didattica quotidiana. Alla base di processi inclusivi di qualità vi sono i percorsi formativi, iniziali e in servizio, per gli insegnanti, nello specifico quelli di sostegno. Questi ultimi da considerare quali reali agenti del cambiamento nella scuola, il cui bagaglio di conoscenze e competenze va oltre un curriculum formativo di soli contenuti. Classi eterogenee composte da alunni con bisogni

molto diversi tra di loro coinvolgono i docenti in processi ciclici di azione e riflessione, nei quali è necessario porsi costanti interrogativi per giungere ad approfondimenti e far fronte alle nuove sfide educative. Il profilo dell'insegnante inclusivo prevede il saper valorizzare la diversità dell'alunno, il lavorare e collaborare con gli altri, riflettere sul proprio ruolo ed operato perseguendo lo sviluppo professionale continuo. In tale ottica, all'interno del percorso universitario dedicato ai futuri docenti specializzati, le ore dedicate alle nuove Tecnologie applicate alla didattica speciale hanno consentito di riflettere ed acquisire consapevolezza su come le TIC possano offrire un notevole contributo, per rinforzare la cultura dell'inclusione, facilitando la predisposizione di percorsi didattici rivolti all'intera classe e permettendo esperienze di apprendimento migliori per tutti. Come emerso nell'esperienza descritta, diventa fondamentale per il docente moderno sapersi orientare nella scelta degli strumenti più adeguati alle diverse situazioni, assumendo di volta in volta il ruolo di tutor, mentore, facilitatore, osservatore dei comportamenti durante le fasi di apprendimento e di co- costruzione del sapere che si intrecciano tra di loro senza soluzione di continuità. In un tale contesto didattico l'utilizzo di strumenti rispondenti ai criteri di accessibilità, usabilità, praticabilità, flessibilità e adattabilità, permette di motivare gli allievi, valorizzare le intelligenze multiple, potenziare l'autostima dei soggetti svantaggiati ed includerli completamente nel gruppo classe, migliorandone di conseguenza le dimensioni della comunicazione e della relazionalità. Nella prospettiva di acquisire un elevato profilo di professionalità, una capacità essenziale dell'insegnante specializzato è possedere un adeguato livello di competenza digitale per saper selezionare e/o produrre documenti e materiali didattici facilmente accessibili per allievi che fruiscono di contenuti in modo non convenzionale, così da garantire a tutti pari opportunità di partecipazione.

Bibliografia

- Akcayir, M., & Akcayir, G. (2017). Advantages and Challenges Associated with Augmented Reality for Education: A Systematic Review of the Literature. *Educational Research Review*, 20, 1-11.
- Alhumaidan, H., Pui Ying Lo, K., & Selby, A. (2018). Co-designing with children a collaborative augmented reality book based on a primary school textbook. *International Journal of Child-Computer Interaction*, (15), 24-36.
- Bauman, Z. (2020). Ripensare il curriculum. Principi educativi e strategie didattiche. Roma: Carocci.
- Béttelheim, B. (1976). Il mondo incantato: uso, importanza e significati delle fiabe. New York: Alfred A. Knopf.
- Bochicchio, F. (2017). L'agire inclusivo della scuola. Logiche, metodologie e tecnologie per insegnanti ed educatori. Lecce: Libellula.
- Calvani, A. (2004). Che cos'è la tecnologia nell'educazione. Roma: Carocci.

- Carlisle, D. (2010, April). graphicx: Enhanced support for graphics. Tratto da <http://www.ctan.org/tex-archive/help/Catalogue/entries/graphicx.html>
- Cottini, L. (2014). Promuovere l'inclusione. L'insegnante specializzato per le attività di sostegno in primo piano. *Italian Journal of Special Education for Inclusion* , 2 (2).
- D'Alonzo, L. (2012). *Come fare per gestire la classe nella pratica didattica*. Firenze: Giunti Scuola
- Deterding, S., Dixon, D., Khaled, R., & Nacke, L. (2011). From Game Design elements to gamefulness: defining "gamification". *Envisioning Future Media Environments*. MindTreck.
- Di Martino, V., & Longo, L. (2019). Realtà aumentata per favorire un apprendimento inclusivo. *Form@re*, XIX, 1 , 179-194.
- Gaspari, P. (2019). L'insegnante specializzato tra passato presente e futuro In P. Sandri, *Rigenerare le radici per fondare i processi inclusivi. Dalla legge 517/77 alle prospettive attuali* (a cura di). Milano: FrancoAngeli.
- Ianes. (2001). Il bisogno di una "speciale normalità" per l'integrazione. *Difficoltà di apprendimento*. 7 (2).
- Ianes, D. (2005). *Bisogni educativi speciali e inclusione. Valutare le reali necessità e attivare tutte le risorse*. Trento: Erickson.
- Kearsley, G. (2000). *Online education: Learning and teaching in cyberspace*. Belmont, CA: Wadsworth.
- Klopfer, E., & Sheldon, j. (2010). Augmenting your own reality: student authoring of science-based augmented reality games. *New Directions for Youth Development* , 128, 85-94.
- Mace, L. (1985). *Universal Design, Barrier Free Environments for Everyone*. Los Angeles: Designers West.
- Milgram, P., & Kishino, F. (1994). A taxonomy of mixed reality visual displays. *IEICE Trans. Information and Systems* , 1321-1329.
- Montanari, M., & Ruzzante, G. (2020). Formare l'insegnante specializzato: l'esperienza inclusiva dei laboratori nel corso di specializzazione per il sostegno. *Italian Journal of Special Education for Inclusion*, V.8 N.1 .
- Muoio, P. (2020). Gioco e apprendimento: spazi, tempi e strumenti nella didattica a distanza. In G. Adorni, A. De Lorenzo, L. Manzoni, & E. Medvet (A cura di), "Smarter School for Smart Cities", atti del Convegno "Didamatica 2020". Università degli Studi di Trieste.
- OMS. (2011). *Rapporto annuale sulla disabilità*. New York. New York.
- Pavone, M. (2004). La formazione degli insegnanti per gestire l'eterogeneità in seno al gruppo classe. *L'integrazione scolastica e sociale* , 3 (2), 127.
- Piu, C. (2007). *Riflessioni di natura didattica*. Roma: Monolite Editrice.
- Roncaglia, G. (2013). L'uso degli e-book nella formazione e nella didattica: tre riflessioni sugli e book di testo e sul loro ruolo. *La vita scolastica* , 67 (10).
- Rose, D., & Meyer, A. (2002). *Teaching Every Student in the Digital Age: Universal Design for Learning*. Alexandria: VA: ASCD.

- Sibilio, M. (2016). *Vicarianza e didattica. Corpo, cognizione, insegnamento*. Brescia: La Scuola Editrice.
- Sural, I. (2018). *Augmented Reality Experience: Initial Perceptions of Higher Education Students*. *International Journal of Instruction*, 11(4) , 565-576.
- UNESCO. (1994). UNESCO (1994), . *The Salamanca Statement and Framework for Action on Special Needs Education* . Salamanca.
- Voronkov, A. (2004). *EasyChair conference system*. Tratto da easychair.org
- Voronkov, A. (2014). *Keynote talk: EasyChair*. In *Proceedings of the 29th ACM/IEEE International Conference on Automated Software Engineering* (p. 3-4). ACM.
- Voronkov, A., & Hoder, K. (s.d.). *Templates*. Tratto da *Templates for proceedings*: <https://easychair.org/proceedings/template.cgi?a=12732737>
- Wikipedia. (s.d.). *EasyChair*. Tratto da Wikipedia: <https://en.wikipedia.org/wiki/EasyChair>
- Zappaterra, T. (2014). *Formare insegnanti specializzati per il sostegno in Italia. Uno sguardo*. Formalex (1).

LaTeX tra competenze digitali e accessibilità: un'esperienza di PCTO con il Laboratorio Polin

**Tiziana Armano¹, Carola Manolino¹, Margherita Piroi¹, Massimo Borsero²,
Davide Maietta¹ e Anna Capietto¹**

¹Dipartimento di Matematica "G. Peano", Università degli studi di Torino

²Istituto Comprensivo "Parri – Vian", Torino

tiziana.armano@unito.it

carola.manolino@unito.it

margherita.piroi@unito.it

massimo.borsero@gmail.com

davide.maietta@unito.it

anna.capietto@unito.it

1

0

Sommario

Il Laboratorio Polin si occupa di ricerca e sviluppo di soluzioni per l'accessibilità di contenuti scientifici per studenti con disabilità e DSA. Nell'ambito della Terza Missione per la diffusione sul territorio di queste soluzioni, durante lo scorso anno scolastico, il Laboratorio ha erogato un MOOC per la formazione insegnanti e un progetto PCTO indirizzato a studenti delle scuole superiori, con lo scopo di diffondere l'utilizzo del linguaggio LaTeX come strumento inclusivo per la scrittura di testi con contenuto matematico. In questo contributo descriviamo peculiarità, obiettivi e svolgimento di questo PCTO, mostrando come un tale progetto possa supportare lo sviluppo di competenze di tipo digitale e sociale.

Introduzione

Il Laboratorio "S. Polin" è un laboratorio di ricerca del Dipartimento di Matematica "G. Peano" dell'Università degli Studi di Torino e si occupa di ricercare soluzioni al problema dell'accesso a studi scientifici da parte di studenti con disabilità e DSA. Oggigiorno, la lettura e la scrittura di testi ben strutturati e "in linea" (ovvero senza grafici o formule) non rappresentano più un problema per persone con disabilità e DSA, soprattutto grazie alla disponibilità di computer o dispositivi mobili e ausili (quali sintesi vocali, display braille e ingranditori). D'altro canto,

1

0

invece, la completa fruizione di risorse didattiche con contenuti STEM (che includono formule, grafici e diagrammi) rimane un problema aperto, nonostante l'attuale notevole sviluppo tecnologico [2].

Per rispondere a queste problematiche, il Laboratorio è impegnato in numerosi progetti di ricerca e sviluppo:

1. **Axessibility** [15]: soluzione per l'accessibilità di testi con formule. È un modulo (pacchetto) dell'ambiente per la scrittura di documenti scientifici in LaTeX. Permette di creare documenti PDF da documenti LaTeX nei quali le formule matematiche possono essere lette mediante le tecnologie assistive per le persone con disabilità visive, quali screen reader e barre braille. È in corso di sviluppo la versione 4.0 che permetterà di ottenere documenti LaTeX PDF totalmente accessibili con formule incorporate, oltre che in LaTeX, in MathML con supporto Mathjax.
2. **Audiofunctions.web** [13]: soluzione inclusiva per l'accessibilità dei grafici di funzione. È una applicazione web che, attraverso tecniche di sonificazione, permette di ottenere grafici di funzione a una variabile reale, esplorabili anche tramite suono. È in corso di sviluppo la versione 2.0 che prevede lo sviluppo di nuove funzionalità e il miglioramento di quelle precedenti, in relazione a una nuova fase di sperimentazione con disabili visivi e con studenti di scuole di diverso ordine e grado.
3. **Novagraphs**: progetto interdipartimentale per lo sviluppo di una soluzione al problema dell'accesso all'informazione scientifica contenuta nelle strutture grafiche (quali tabelle, diagrammi entità-relazione, diagrammi UML, alberi, grafi, circuiti...) da parte di persone con disabilità visive. Prevede lo sviluppo di una soluzione software che produce in modo semi-automatico descrizioni testuali navigabili in modo interattivo con comando vocale di strutture grafiche complesse.
4. **VoiceMath**: soluzione per l'accessibilità di video di lezioni con formule. Il video di una lezione (risorsa didattica molto diffusa nel periodo pandemico) presenta diversi problemi di accessibilità: necessita di sottotitoli adeguati per persone con disabilità uditive, della trascrizione dell'audio per persone con disabilità motorie, visive e con DSA che hanno difficoltà a prendere appunti, e di un documento digitale accessibile dei contenuti video. Esistono attualmente numerosi sistemi per la trascrizione e la sottotitolazione automatica, ma nessuno gestisce le formule in modo opportuno: per la fruizione di un video con contenuti scientifici serve la trascrizione di formule in linguaggio specifico (LaTeX o MathML). VoiceMath, finanziato da fondazione CRT, Dipartimento di Matematica "G. Peano" e Direzione SIPE (UniTO), è una soluzione software per la trascrizione in testo + LaTeX di lezioni universitarie "con formule", utile a persone con disabilità motorie, uditive e con DSA. Il software è stato realizzato in collaborazione con H-Farm Innovation ed è attualmente in corso di brevetto in UniTO e sarà disponibile per i docenti dell'Università di Torino da novembre 2022.

5. SpeechMate: il progetto prevede lo sviluppo di una soluzione software per dettatura e modifica di formule matematiche basata su funzionalità di interazione vocale con riscontro visivo e vocale dei contenuti dettati, utile a persone con disabilità motorie, visive o DSA, nella scrittura, nell'elaborazione e nella risoluzione di espressioni matematiche. Le persone con disabilità o disordini motori agli arti superiori o disabilità visive incontrano difficoltà nella scrittura della matematica e l'esecuzione di semplificazioni numeriche o simboliche di espressioni matematiche è complessa per persone con DSA. Attualmente non sono disponibili software con funzionalità di dettatura e modifica di formule in lingua italiana, esistono solo alcune parziali soluzioni in lingua inglese. Il Laboratorio ha sviluppato un prototipo del software partendo dall'esperienza di SpeechMatE 0.1, un software con interazione multimodale stabile, con funzioni di dettatura anche di matematica avanzata, di navigazione e modifica delle formule.

Oltre a ricerca e sviluppo, il Laboratorio Polin ha come obiettivo primario la disseminazione sul territorio delle soluzioni per l'accessibilità delle STEM e della cultura dell'accessibilità in generale. In questo ambito il Laboratorio Polin ha erogato corsi di formazione per insegnanti curricolari e di sostegno e ha proposto attività di diffusione della cultura dell'accessibilità a studenti di scuole secondarie. In particolare, nell'inverno 2022, ha proposto il MOOC gratuito Accessibilità delle STEM: pratiche didattiche e tecnologiche per non vedenti, per fornire agli insegnanti strumenti e pratiche didattiche per l'insegnamento della matematica e della fisica per studenti con disabilità visive. Sempre nell'ambito della Terza Missione nell'estate 2022 è stato realizzato un progetto PCTO (Percorsi per le Competenze Trasversali e l'Orientamento) "Studenti e studentesse ambasciatori della matematica accessibile". Tale progetto, oggetto di questo contributo, rientra tra i progetti di "alternanza scuola-lavoro" proposti dall'Università degli Studi di Torino.

2 Peculiarità del progetto

Il progetto ha visto la partecipazione di 57 studenti: 16 ragazze e 41 ragazzi, di 8 diverse scuole secondarie di secondo grado di Torino e provincia, e di Cuneo. Nello specifico 1 partecipante aveva terminato la classe prima, 1 la classe seconda, 6 la classe quarta, e i rimanenti la classe terza. Per quanto riguarda gli indirizzi di studio, gli studenti provenivano da liceo classico, liceo scientifico (tradizionale e scienze applicate), istituto tecnico settore economico, e istituto tecnico settore informatico, come descritto in Tabella 1.

Indirizzo	classe I	classe II	classe III	classe IV	tot
Liceo classico	1	1	-	-	2
L. scientifico trad.	-	-	9	2	11
L. scientifico scienze applicate	-	-	9	-	9
IT economico	-	-	1	4	5
IT informatico	-	-	30	-	30
tot	1	1	49	6	57

Tabella 1

Il progetto ha riscosso un notevole interesse, tanto che numerose richieste sono state respinte essendo già stato raggiunto il tetto massimo di studenti partecipanti. Crediamo che questo forte interesse sia stato motivato dal duplice valore formativo del progetto. Esso infatti aveva come scopo:

1. sviluppare le competenze digitali e il pensiero computazionale degli studenti;
2. supportare le loro competenze sociali, promuovendo la cultura dell'inclusione e dell'accessibilità.

L'obiettivo primario del progetto era quello di introdurre gli studenti all'uso di LaTeX, il quale non solo è il linguaggio più diffuso per la stesura di testi scientifici, ma risulta anche un'ottima soluzione ai problemi di accessibilità, sia per la lettura che per la scrittura di testi contenenti formule da parte di disabili visivi [3]. A differenza dei classici word-processor che si basano sul paradigma WYSIWYG (What You See Is What You Get), "ciò che vedi è ciò che ottieni", LaTeX è un text processor e permette di lavorare in modalità WYSIWYM (What you See Is What You Mean), "ciò che vedi è ciò che vuoi dire". Questo significa che chi scrive in LaTeX deve concentrarsi sul contenuto e sulla struttura del testo che vuole produrre, tramite la scrittura di un codice, mentre gli aspetti tipografici vengono controllati autonomamente dal programma. Per questo, la scrittura con questo linguaggio di marcatura permette di sviluppare il pensiero computazionale. Tale competenza è esplicitamente richiamata nelle Linee Guida per il passaggio al nuovo ordinamento per secondo biennio e quinto anno degli istituti tecnici, come anche nelle Indicazioni Nazionali e in particolare tra gli obiettivi specifici di apprendimento per i licei [6, 9].

Parallelamente, il corso intendeva supportare le competenze sociali degli studenti, favorendo lo sviluppo di conoscenze in tema di inclusività. L'importanza del potenziamento della cultura dell'inclusione, oltre a essere già messa in rilievo dalla Legge Quadro 104/92 [7] è stata sottolineata anche più recentemente dalla legge 170/10 [10] e successive note applicative. Inoltre, anche le stesse Linee Guida per i Percorsi per le Competenze Trasversali e per l'Orientamento affermano l'importanza di "elevare i livelli di cittadinanza attiva, di coesione e protezione sociale, favorendo la partecipazione, l'inclusione e il pieno sviluppo della persona, a valorizzare il potenziale di crescita e di occupazione lavorativa, in attuazione degli articoli 2, 3, 4, 9, 18 e 118, quarto comma, della Costituzione" [8, p. 18].

È stato introdotto il pacchetto Accessibility, così da presentare fin da subito LaTeX agli studenti come linguaggio di marcatura utile e vantaggioso, non solo per la sua praticità, gratuità e flessibilità [3,4], ma anche per l'opportunità che offre di redigere testi scientifici accessibili.

3 Descrizione delle attività realizzate

Il progetto si è sviluppato tra il 13 giugno 2022 e il 25 luglio 2022, in 30 ore totali, di cui 8 di corso introduttivo a LaTeX (suddivise in due lezioni frontali da 4 ore ciascuna), 20 di lavoro degli studenti in autonomia da remoto, e ultime 2 ore di restituzione in presenza dei lavori realizzati dagli studenti.

Le due lezioni frontali si sono svolte presso uno dei laboratori informatici del Dipartimento di Matematica dell'Università di Torino, dove ha sede il Laboratorio Polin. Si è voluto introdurre il corso proprio partendo da riflessioni sul tema dell'accessibilità, per sottolineare sin dall'inizio la centralità di questa tematica nell'intero PCTO. Gli studenti hanno avuto modo di visitare il laboratorio, dove sono stati mostrati e descritti i principali strumenti utilizzati per la creazione di materiale didattico accessibile, come il fornetto (Figura 1), la stampante 3D (Figura 2) e la stampante a rilievo (Figura 3a e 3b).



Figura 1



Figura 2



Figura 3a



Figura 3b

È stata inoltre presentata agli studenti l'applicazione AudioFunction.web [13], per dare un'idea delle soluzioni innovative sviluppate dal laboratorio per l'accessibilità dei contenuti matematici. Ha completato questa parte introduttiva una testimonianza diretta di una collaboratrice del Laboratorio, la Dott.ssa Ester Tornavacca, laureata in matematica e ipovedente, che ha presentato la propria esperienza in quanto studentessa e lavoratrice in ambito scientifico con disabilità visiva. Successivamente, sono state presentate agli studenti le principali caratteristiche e funzionalità di LaTeX: preambolo e struttura del testo principale, funzionalità e sintassi dei comandi, inserimento di formule in modalità "in corpo"

e “fuori corpo” [14]. Tali caratteristiche sono state insegnate in maniera indiretta. I tutor del corso hanno fornito alcuni esempi di documenti pdf redatti con LaTeX e relativi codici e gli studenti, lavorando a piccoli gruppi, dovevano provare a identificare i principali elementi del codice e capirne struttura e funzionalità. Essendo fisicamente presenti in un laboratorio informatico, avevano modo di testare immediatamente le proprie congetture a riguardo. Tale lavoro era intervallato da momenti di discussione collettiva in cui venivano esposte, commentate ed eventualmente validate le osservazioni emerse. La scelta di non insegnare mostrando e spiegando ogni step del processo di realizzazione di un documento con LaTeX, ma permettendo agli studenti di approcciarsi a questo linguaggio apprendendo “per scoperta”, è dovuta al fatto che nei progetti PCTO la figura del tutor “supporta e favorisce i processi di apprendimento dello studente. Il tutor si connota come ‘facilitatore dell’apprendimento’; accoglie e sostiene lo studente nella costruzione delle proprie conoscenze; lo affianca nelle situazioni reali e lo aiuta a ri-leggere l’insieme delle esperienze per poterle comprendere nella loro naturale complessità. Lo aiuta, dunque, a rivisitare il suo sapere e ad avere chiara valutazione delle tappe del proprio processo di apprendimento” [9, p. 15]. Questa concezione della figura dei tutor ha guidato la progettazione dei loro compiti e interventi anche nelle fasi successive del progetto, in cui erano previsti lavoro autonomo dei gruppi di studenti e restituzione collettiva.

La parte centrale del progetto si è infatti svolta da remoto e il lavoro è stato coordinato dai tutor tramite piattaforma Moodle. Gli studenti hanno autonomamente formato dei gruppi di lavoro da 2 o 3 membri, e a ciascun gruppo è stato richiesto di trascrivere in LaTeX una parte del libro “Matematica come scoperta” di Giovanni Prodi [5]. Questo libro in due volumi, la cui ultima edizione di stampa risale al 1975, è un importante testo di riferimento per la Didattica della Matematica. La scelta di far trascrivere questo intero libro è stata motivata dal desiderio che il progetto potesse anche portare a un output utile, come la realizzazione della versione digitale accessibile di un così interessante libro di testo. Come già descritto, per esempio in [3], la peculiarità di LaTeX, rispetto ad altre soluzioni per l’accessibilità di testi matematici per persone con disabilità visiva, è essere pienamente inclusivo. La versione digitale e accessibile di questo libro di testo può essere infatti una risorsa assolutamente utile anche per persone senza alcuna disabilità visiva.

3.1 Lavoro svolto dagli studenti in autonomia

A partire da quanto appreso nel corso in presenza (funzionalità principali di LaTeX e codice per scrivere espressioni matematiche basilari), i membri di ciascun gruppo hanno dovuto cercare in autonomia soluzioni per la trascrizione del testo: ad esempio, i comandi per scrivere il testo in corsivo o grassetto, oppure per creare elenchi puntati, inserire immagini, o ancora inserire espressioni o simboli matematici più complessi. Su Moodle era a disposizione un forum che poteva essere utilizzato dagli studenti per scambiarsi domande e informazioni reperite, e dove eventualmente interpellare i tutor, i quali potevano rispondere a dubbi o

questioni. Inoltre, una volta a settimana (per un totale di 5 incontri) era previsto un incontro online con i tutor in cui potevano essere risolti ulteriori problemi incontrati dagli studenti.

L'utilizzo di una piattaforma online per lo scambio di informazioni e la gestione del lavoro, e la necessità di ricavare autonomamente informazioni riguardo l'uso di LaTeX (quindi il dover imparare a cercare un'informazione specifica, a riconoscere una fonte affidabile o a conoscere il funzionamento di forum di informatica), hanno contribuito al raggiungimento del primo obiettivo del progetto. La competenza digitale, come si legge nella Raccomandazione del Parlamento europeo del 2006, infatti non consiste solo nel saper utilizzare le tecnologie della società dell'informazione – in questo caso saper scrivere un testo col linguaggio LaTeX –, ma “essa è supportata da abilità di base nelle TIC: l'uso del computer per reperire, valutare, conservare, produrre, presentare e scambiare informazioni nonché per comunicare e partecipare a reti collaborative tramite Internet” [11, p. 15]. Questo approccio, inoltre, supporta lo sviluppo della competenza personale, sociale e la capacità di imparare a imparare – competenza chiave nel quadro definito dalla Raccomandazione del Consiglio europeo del 2018 –, che consiste “nella capacità di riflettere su sé stessi, di gestire efficacemente il tempo e le informazioni, di lavorare con gli altri in maniera costruttiva, di mantenersi resilienti e di gestire il proprio apprendimento e la propria carriera” [12, p. 10].

3.2 Restituzione collettiva del lavoro svolto

A un mese dall'assegnazione del lavoro a gruppi, tutor e studenti si sono ritrovati in presenza per una restituzione del lavoro svolto. Gli studenti hanno esposto eventuali difficoltà riscontrate nell'uso di LaTeX, e i tutor hanno condiviso con l'intero gruppo osservazioni già fatte ai singoli gruppi, o durante le sessioni di tutorato online, e problematiche emerse nel forum con relative soluzioni.

Globalmente, gli studenti hanno mostrato un notevole grado di autonomia nello svolgimento della trascrizione e nella risoluzione di problemi relativi all'uso di LaTeX. I principali dubbi sollevati nel forum e tutorati online, e difficoltà emerse dalle correzioni delle trascrizioni, riguardavano principalmente:

- l'inserimento di immagini;
- la sintassi per l'inserimento di formule matematiche.

Riguardo a quest'ultimo punto, la maggior parte degli studenti tendeva a segmentare in più parti una singola formula. Ad esempio, per trascrivere la parte di testo in Figura 4a è stato utilizzato il codice in Figura 4b. In figura 4a vediamo l'uso del simbolo del dollaro in apertura e chiusura di ogni parte di un'unica formula, la cui scrittura risulta così appesantita. La stessa poteva invece essere scritta come:

$$\overline{PQ} = \overline{PT} + \overline{TQ} = |x_2 - x_1| + |y_2 - y_1|$$

(Non ci soffermiamo qui su ulteriori errori fatti nella scrittura di questo codice).

$$\overline{PQ} = \overline{PT} + \overline{TQ} = \sqrt{x_2 - x_1} + \sqrt{y_2 - y_1}$$

D'altra parte si ha $\sqrt{x_2 - x_1} = \sqrt{(x_2 - x_1)}$ perché $\sqrt{x_2 - x_1}$ può essere uguale a $x_2 - x_1$, oppure a $x_1 - x_2 = (-1)(x_2 - x_1)$: passando al quadrato il fattore (-1) si trasforma in 1. Questa è la formula che esprime la distanza tra P e Q mediante le coordinate. È una formula di grande interesse: essa permette intanto di descrivere in termini algebrici una figura molto importante: il cerchio.

Figura 4a

$$\overline{PQ} = \overline{PT} + \overline{TQ} = \sqrt{x_2 - x_1} + \sqrt{y_2 - y_1}$$

D'altra parte si ha $\sqrt{x_2 - x_1} = \sqrt{(x_2 - x_1)}$ perché $\sqrt{x_2 - x_1}$ può essere uguale a $x_2 - x_1$, oppure a $x_1 - x_2 = (-1)(x_2 - x_1)$: passando al quadrato il fattore (-1) si trasforma in 1. Questa è la formula che esprime la distanza tra P e Q mediante le coordinate. È una formula di grande interesse: essa permette intanto di descrivere in termini algebrici una figura molto importante: il cerchio.

Figura 4b

Sono invece quasi tutti riusciti in autonomia a capire come:

- inserire tabelle;
- usare caratteri speciali, come grassetto, corsivo, ecc. e cambiare le dimensioni del testo;
- inserire espressioni matematiche complesse, come quelle in figura 5.

$$c^2 = \overline{AB^2} = (b \cos \gamma - a)^2 + (b \sin \gamma)^2 = b^2(\cos \gamma)^2 - 2ab \cos \gamma + a^2 + b^2(\sin \gamma)^2 = a^2 - 2ab \cos \gamma + b^2[(\cos \gamma)^2 + (\sin \gamma)^2]$$

Figura 5

Alcuni dei testi prodotti sono anche stati testati con lo screen reader NVDA, per verificarne l'effettiva accessibilità. Questo momento di restituzione è stato anche occasione per mostrare alcune caratteristiche del pacchetto Aaccessibility, come la necessità di usare specifici dizionari [1].

4 Feedback degli studenti sul progetto

A conclusione del progetto, è stato somministrato a tutti i partecipanti un questionario online di valutazione dell'esperienza. Dalle risposte date dagli

studenti ai vari quesiti abbiamo potuto constatare il raggiungimento di tutti gli obiettivi preposti alla progettazione del progetto.

Su un piano più generale abbiamo riscontrato entusiasmo da parte degli studenti per la strutturazione del PCTO. Molti hanno manifestato stupore a partire dal confronto con attività pregresse. Una studentessa a riguardo afferma:

“Visto le esperienze in altri laboratori PCTO non mi aspettavo grandi cose; nello specifico, non mi aspettavo di imparare qualcosa di nuovo a livello più pratico”.

Numerosi studenti hanno sottolineato l'entusiasmo per la partecipazione ad attività in presenza e la possibilità di lavorare in gruppo tra pari, mettendo in luce il bisogno di essere coinvolti in attività presenziali e collaborative. Sicuramente questa è un'esigenza ancora più sentita in questo periodo storico di situazione post-pandemica delle nostre scuole.

A conferma dell'importanza della missione di orientamento all'interno dei progetti PCTO, si è dimostrata vincente anche la scelta di realizzare le lezioni in presenza nelle aule del Dipartimento di Matematica. A riguardo per esempio due studenti affermano:

“La cosa che mi è piaciuta di più è entrare all'università per la prima volta e seguire le lezioni nelle aule”.

“Per essere stata la prima edizione, mi è proprio piaciuto, anche dato il grande impegno che ho davvero percepito da parte di tutti gli organizzatori. In più la sola esperienza di entrare in dei locali universitari ed assistere ad un vero corso a tutti gli effetti è stata davvero particolare ed interessante per uno solamente al terzo anno di superiori”.

Riguardo al primo obiettivo specifico delineato è evidente l'apprezzamento da parte di tutti i partecipanti per le competenze digitali maturate; in particolare molti studenti fanno a riferimento all'utilità pratica che il linguaggio LaTeX potrà avere nella loro vita, dichiarando ad esempio:

“Aver appreso come utilizzare LaTeX mi sarà utile per formattare testi, specialmente dal momento che ho intenzione di intraprendere una carriera universitaria ad indirizzo scientifico”

É evidente anche la consapevolezza da parte di molti partecipanti di come le nuove conoscenze e abilità apprese contribuiscano in modo più ampio allo sviluppo della competenza digitale e del pensiero computazionale. Lo intuiamo per esempio da risposte come:

“In questo corso ho imparato a scrivere usando il linguaggio LaTeX, con la quale si possono creare documenti matematici che sono accessibili a non vedenti/ipovedenti (abbiamo anche imparato la differenza tra non vedente e ipovedente), incrementando la mia destrezza nell'uso del computer e imparando anche ad essere un po' più autonomo”.

“A mio parere, LaTeX è complesso, perché richiede un approccio diverso al modo di scrivere. Con Word e OpenOffice è facile scrivere subito, perché si pensa alla forma, mentre con latex si pensa al contenuto. Secondo me per imparare LaTeX si deve andare per gradi”.

In quest'ultima risposta, per esempio, è evidente la consapevolezza della differenza tra la scrittura con un word-processor WYSIWYG (lo studente qui fa riferimento a Word e OpenOffice) e un text processor WYSIWYM, come LaTeX.

Quando nel questionario viene chiesto quali siano le competenze apprese durante il PCTO, quasi la totalità dei partecipanti fa riferimento sia alle competenze digitali, sia a competenze in materia di inclusione. Le risposte mettono in luce come, al termine del percorso, gli studenti fossero sensibili al tema della disabilità visiva, riconoscendo anche la propria responsabilità nel favorire processi di inclusione e l'importanza dello sviluppo di risorse per l'accessibilità. Riportiamo alcune risposte significative a questo riguardo:

“Direi di iscriversi perché si parla di argomenti che non vengono affrontati da altre parti, per esempio io non avevo mai sentito parlare di ipovedente.”

“Ho imparato a scrivere in LaTeX, che è un linguaggio di marcatura per la scrittura di testi scientifici e matematici. Oltre ad aver acquisito tali competenze, credo che il messaggio da trasmettere è che LaTeX mira ad includere e rendere accessibile ogni tipo di testo per tutti, soprattutto per le persone con disabilità visiva.”

“Ho imparato come trascrivere testi con LaTeX, ho acquisito competenze nel lavorare in un gruppo ed ho potuto rendermi conto di quanto la strada verso la completa accessibilità sia ancora lontana, e del fatto che è anche nostro compito renderla più facile da percorrere.”

5 Conclusioni

L'insegnamento del LaTeX in ottica inclusiva si è dimostrato essere adatto per un progetto PCTO che coinvolga e formi gli studenti su diversi fronti:

- l'educazione informatica e lo sviluppo del pensiero computazionale;
- l'educazione all'inclusività;
- lo sviluppo della competenza “personale, sociale e la capacità di imparare a imparare”.

In primis, gli studenti hanno avuto l'opportunità, altrimenti ancora rara in contesto scolastico, di apprendere il linguaggio LaTeX. Questa è una competenza assolutamente utile per il futuro universitario e lavorativo degli studenti, che se ne sono mostrati consapevoli, come emerso dalle loro risposte al questionario. LaTeX non solo è il linguaggio più usato per scrivere testi scientifici, ma consente di scrivere testi di qualsiasi genere, garantendone un'impaginazione e una grafica di alta qualità. Non bisogna poi dimenticare che può essere utilizzato in maniera gratuita e accessibile su qualsiasi sistema operativo. Dal punto di vista dell'educazione informatica, la conoscenza del linguaggio LaTeX non è stato

l'unico risultato ricercato ed ottenuto. Il processo stesso dell'apprendimento di LaTeX, così come impostato nel corso e nelle attività proposte, ha contribuito all'obiettivo di promuovere la competenza digitale.

Aspetto peculiare del corso è aver avvicinato gli studenti al tema dell'accessibilità, inserito nella questione più ampia dell'inclusione. Sebbene, quasi inevitabilmente, nella seconda parte gli studenti si siano concentrati maggiormente sull'apprendimento del LaTeX, col rischio di dimenticare le questioni relative all'accessibilità e inclusività, l'incontro finale in presenza ha voluto proprio rimettere l'accento su questi aspetti. Sebbene anche nelle risposte ai questionari sembra che gli studenti abbiano apprezzato maggiormente l'opportunità di apprendere LaTeX, non sono mancate risposte che hanno mostrato come anche il tema dell'inclusione abbia riscosso interesse. Particolarmente utile in questa prospettiva di educazione all'inclusione è risultata essere la testimonianza diretta della nostra collaboratrice, che apprezziamo tramite le parole di una studentessa:

“È stata un'esperienza che ci permette di aprire gli occhi anche su altre prospettive, dovremmo essere una società in grado di poter includere tutti allo stesso modo e dare a tutti le stesse possibilità.”

Infine, le modalità di svolgimento del corso, la scelta di far apprendere i contenuti non in maniera diretta ma “per scoperta”, l'organizzazione del lavoro in modalità ibrida e facendo lavorare gli studenti in gruppi, l'invito a visitare e lavorare nei locali universitari, hanno favorito certamente lo sviluppo di competenze trasversali come la “capacità di gestire efficacemente il tempo e le informazioni”, “capacità di imparare e di lavorare sia in modalità collaborativa sia in maniera autonoma”, “capacità di lavorare con gli altri in maniera costruttiva”, “capacità di comunicare costruttivamente in ambienti diversi”, “capacità di pensiero critico e abilità integrate nella soluzione dei problemi” e “curiosità nei confronti del mondo, apertura per immaginare nuove possibilità” [8].

Quella descritta è stata la prima edizione di un progetto che intendiamo riproporre e migliorare, anche e soprattutto grazie ai feedback ricevuti dagli studenti che hanno partecipato. Ad esempio, da molti studenti è stato manifestato il desiderio di apprendere ad utilizzare software di sintesi vocale come NVDA, anche per poter testare in autonomia l'accessibilità delle risorse digitali prodotte. Questo suggerimento, come altri, saranno presi in considerazione dal Laboratorio Polin per continuare a coinvolgere le scuole e proporre attività di formazione con lo scopo di diffondere la cultura dell'accessibilità delle materie STEM.

Bibliografia e sitografia

1. Ahmetovic, D., Armano, T., Bernareggi, C., Berra, M., Capietto, A., Coriasco, S., Murru, N., Ruighi, A. & Taranto, E. (2018). Axxessibility: a LaTeX Package for Mathematical Formulae Accessibility in PDF Documents.

Proceedings of the 20th International ACM SIGACCESS Conference on Computers and Accessibility ASSETS 2018 (pp. 352-354).

2. Armano, T., Capietto, A., Ahmetovic, D., Bernareggi, C., Coriasco, S., Ducci, M., Magosso, C., Mazzei, A., Muuru, N. & Sodina, A. (2020). Accessibilità di contenuti digitali per le STEM: un problema aperto. Alcune soluzioni inclusive per l'accessibilità di formule e grafici per persone con disabilità e DSA. In Atti del convegno DIDAMATICA 2020 (pp. 2-11).
3. Borsero, M., Murru, D. & Ruighi, A. (2016). Il LaTeX come soluzione al problema dell'accesso a testi con formule da parte di disabili visivi. *Ars Texnica*, Vol. 22, p. 12-18.
4. Giacomini, R. (2016). LaTeX tra i banchi. Possibili applicazioni in ambito scolastico di LaTeX. *Ars Texnica*, Vol. 22, p. 7-11.
5. Prodi, G. (1975). *Matematica come scoperta*. Casa editrice G. D'Anna.
6. Indicazioni nazionali per i Licei. D.P.R. 89 del 15 marzo 2010.
7. Legge-quadro per l'assistenza, l'integrazione sociale e i diritti delle persone handicappate (104/92). *Gazzetta ufficiale della Repubblica italiana*, 39.
8. Linee guida dei percorsi per le competenze trasversali e per l'orientamento (PCTO). Decreto ministeriale 774 del 4 settembre 2019.
9. Linee guida per il passaggio al nuovo ordinamento, secondo biennio e quinto anno, istituti tecnici. Allegato D.P.R. 15 Marzo 2010, articolo 8, comma 3.
10. Nuove norme in materia di disturbi specifici di apprendimento in ambito scolastico (170/10). *Gazzetta ufficiale della Repubblica italiana*, 244.
11. Raccomandazione del parlamento europeo e del consiglio del 18 dicembre 2006 relativa a competenze chiave per l'apprendimento permanente (2006/962/ce).
12. Raccomandazione del Consiglio del 22 maggio 2018 relativa alle competenze chiave per l'apprendimento permanente. (2018/C 189/01).
13. <http://www.integr-abile.unito.it/audiofunctions.web/>
14. <https://www.didatticainterattiva.it/files/LaTeX-facile.pdf>
15. <http://www.integr-abile.unito.it/axessibility/>

PythagorHub: quando Pitagora incontra Android

Paola Palestini¹, Luciano Zazzetti¹

¹ Liceo Scientifico "B. Rosetti", San Benedetto del Tronto, Italia

paola.palestini@liceorosetti.edu.it

luciano.zazzetti@liceorosetti.edu.it

1

0

Sommario

Il progetto qui descritto è stato sviluppato nell'ultima parte dell'anno scolastico 2021-2022, in una classe seconda di un Liceo Scientifico Matematico, all'interno di un laboratorio didattico delle ore aggiuntive che caratterizzano questo tipo di corso.

È stata sviluppata un'app Android che consente di visualizzare delle dimostrazioni di geometria sintetica in modo non tradizionale.

Il tema scelto è stato quello del teorema di Pitagora, per il quale esiste sì una grande quantità di dimostrazioni facilmente reperibili in rete, ma talvolta di comprensione e rielaborazione non immediata. L'esigenza di rivedere il modo di presentare le dimostrazioni di geometria è nata durante la DaD, dall'esigenza di rendere la lezione a distanza più coinvolgente ed efficace, avendo a disposizione quasi illimitate risorse digitali. Coniugare classici argomenti di matematica, rielaborarli in modo accattivante ed inserirli in un contesto tecnologico avanzato ha contribuito a rafforzare la motivazione dello studio della matematica così come la consapevolezza dei processi che stanno dietro al funzionamento di uno smartphone.

1 Introduzione

Nel periodo di chiusura delle aule per l'emergenza sanitaria legata alla pandemia, strumenti e servizi digitali sono stati indispensabili per la gestione del problema della continuità didattica. Scuola e famiglie hanno collaborato per far sì che docenti e discenti potessero avere a disposizione gli strumenti necessari per la connessione. Accanto ai problemi di natura tecnica, legati alla disponibilità dei dispositivi ed ampiezza di banda funzionali al collegamento, la scuola ha dovuto gestire questioni non meno rilevanti quali quelle legate alle metodologie della

1

0

didattica a distanza (DaD). Fare scuola attraverso uno schermo richiede infatti approcci diversi da quelli che si utilizzano nel caso della lezione in presenza e la ricerca di strategie funzionali al coinvolgimento e all'attiva partecipazione degli studenti è stata sicuramente l'aspetto più impegnativo della sfida che i docenti hanno dovuto affrontare in questo periodo. L'aspetto positivo della questione è che le nuove esperienze digitali e didattiche maturate in questo periodo dalla scuola costituiscono ora un bagaglio che può essere speso per innovare anche la didattica in presenza.

Quella qui descritta è un esempio di didattica digitale messa in atto proprio sulla scia delle esperienze acquisite relativamente all'apprendimento della geometria nel periodo della DaD. Questa attività, rivolta ad una classe seconda del Liceo Scientifico Matematico, ha avuto come prodotto finale un'app sulle varie dimostrazioni del teorema di Pitagora. È stata programmata e curata dai due autori del presente articolo i quali, in momenti diversi, hanno gestito rispettivamente parti complementari dello stesso progetto. L'aver affrontato questo percorso nelle ore aggiuntive del Liceo Matematico ha permesso di gestire l'attività con una metodologia di tipo laboratoriale e con tempi distesi: al progetto complessivamente sono state dedicate 20 ore.

2 Un modo diverso di proporre dimostrazioni

Questo laboratorio didattico, che ha richiesto un alto livello di coinvolgimento da parte degli studenti, è stato strutturato secondo una tecnica a puzzle, in cui ognuno ha dovuto curare una sua specifica parte nella realizzazione del tutto. L'obiettivo di ognuno degli studenti è stato quello di trovare il modo di spiegare una particolare dimostrazione del teorema di Pitagora, utilizzando un metodo rigoroso e al tempo stesso efficace e accattivante. Nella maggior parte dei casi gli studenti si trovano nella condizione di riferire una dimostrazione con lo scopo di mostrare a qualcuno, il docente, di averla compresa. In questo caso è stato proposto loro di comprendere la dimostrazione e trovare un modo utile per spiegarla così da favorire la comprensione da parte di chi non la conosce.

Le fasi dell'attività proposta sono quelle qui di seguito elencate.

- Selezione della versione della dimostrazione del teorema su cui incentrare il proprio lavoro. La quasi totalità degli studenti ha concentrato la propria ricerca sul web [1]. La maggior parte di loro ha scelto di attingere a dimostrazioni grafiche e animate, generalmente non corredate da rigorose spiegazioni [2-6].
- Realizzazione di un breve video contenente la descrizione orale della versione della dimostrazione rivista dallo studente. Al fine di rendere la dimostrazione chiara e semplice, oltre che rigorosa, è stato suggerito di affiancare la descrizione a figure realizzate con pennarelli colorati (fig. 1) piuttosto che di figure fatte mediante tavolette grafiche o strumenti affini. I video prodotti sono stati condivisi con il resto della classe.

- Revisione della dimostrazione alla luce dei contributi del docente e degli studenti che hanno preso visione del video.

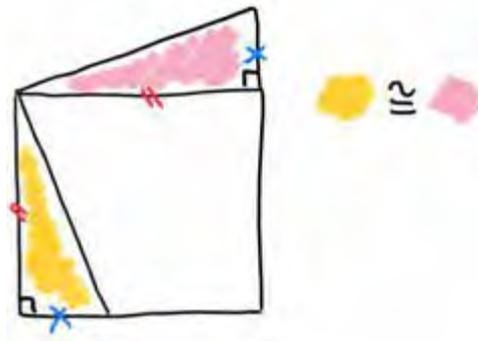


Figura 1

Disegno realizzato da uno studente per spiegare la congruenza fra i triangoli giallo e rosa in figura facendo uso solo di simboli, immagini e colori

- Realizzazione di un breve video che spieghi la dimostrazione mediante un'animazione priva di audio, realizzata mediante un software opportuno. Gli studenti avevano già una discreta conoscenza delle funzionalità di base di GeoGebra 2D e pertanto il docente, premessa la spiegazione di nuove specifiche funzionalità, ne ha suggerito l'uso. Per realizzare l'animazione è stato consigliato di impostare tutta la costruzione su un solo slider. Ad esempio, considerando la dimostrazione a cui si fa riferimento nelle immagini successive, per $0 < d <= 1$, viene ruotato il triangolo rettangolo intorno ad uno dei suoi vertici e la rotazione massima è di 90° (fig. 2). Per $1 < d < 1.5$ viene ultimata la costruzione (fig. 3) e in questo caso, mediante

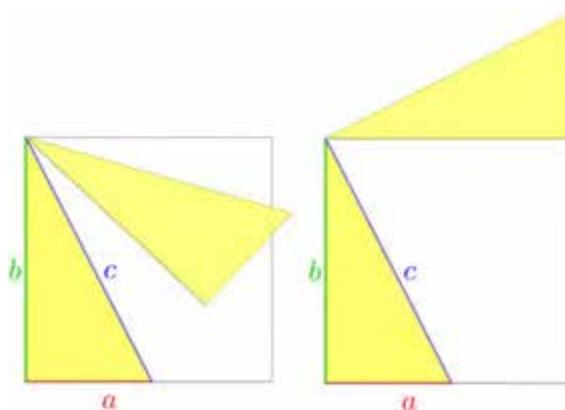


Figura 2

Rotazione del triangolo rettangolo intorno ad uno dei suoi vertici, per $0 < d < 1.5$

l'attribuzione di condizioni agli oggetti, questi vengono visualizzati o nascosti in corrispondenza di determinati valori o intervalli dello slider. La parte testuale, ridotta al minimo, è stata usata talvolta per mettere in evidenza i passaggi algebrici della dimostrazione.

Per la realizzazione del video della costruzione è stato inserito nella finestra *Algebra* [7] il seguente comando
GeoGebraScript:EsportaImmagine("filename","dimostrazione.webm","type","webm","slider",d,"time",33,"scale",2).

Per creare il video basta aprire il file dal browser, ridurre la finestra alla grandezza dell'area da esportare, e cliccare sull'istruzione. La finestra rimane bloccata per il tempo necessario al programma per salvare il video che, in formato WebM, viene poi salvato nella cartella Downloads.

Attingendo a competenze apprese in maniera autonoma, alcuni studenti invece hanno realizzato l'animazione utilizzando la libreria Manim di Python o specifici

software di grafica come Adobe After Effects.

- Scrittura della versione testuale della dimostrazione.

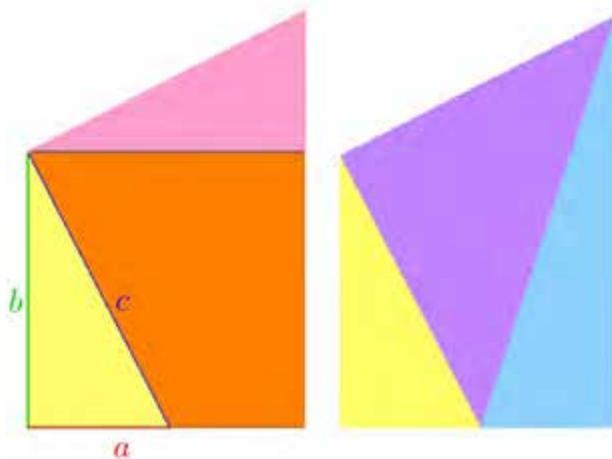


Figura 3
Per $1 < d < 1.5$ viene ultimata la costruzione

In un secondo momento le dimostrazioni sono state riscritte in LaTeX utilizzando lo stile reso famoso da Oliver Byrne nel suo classico "Elementi di Euclide" [8], facendo ricorso a simboli e immagini colorate inserite nel testo in modo da agevolare il processo di comprensione (fig. 4).

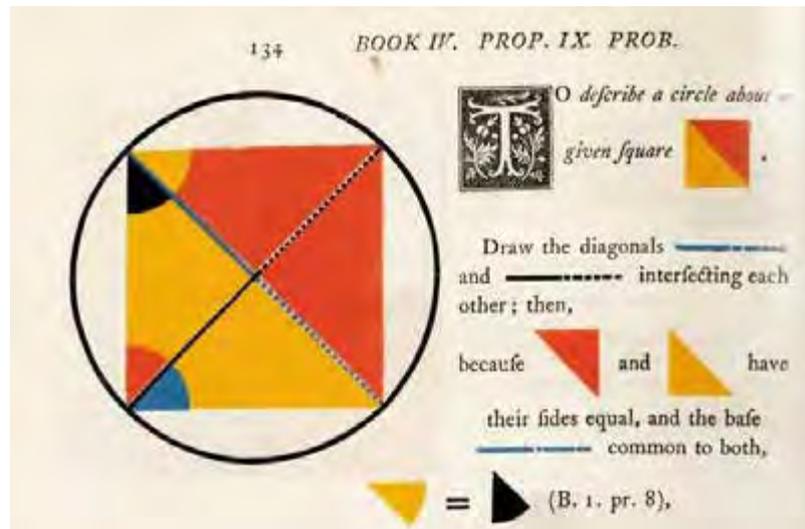


Figura 4

Immagine tratta da "The Elements of Euclid" di Oliver Byrne

- Scrittura della versione testuale della propria versione della dimostrazione nella forma tradizionale.

Successivamente la classe ha collaborato con il docente alla creazione dell'app nella quale sono stati inseriti video e PDF delle dimostrazioni nella versione "Byrne".

3 PythagorHub

Le dimostrazioni, in formato sia statico che dinamico, sono state inserite all'interno di un'app Android realizzata utilizzando l'ambiente di sviluppo App Inventor.

3.1 Sviluppare con App Inventor

MIT App Inventor [9] è un ambiente di sviluppo integrato di applicazioni web originariamente fornito da Google e ora gestito dal Massachusetts Institute of Technology (MIT).

L'applicazione fu rilasciata pubblicamente il 15 dicembre 2010. Nella seconda metà del 2011, Google ha rilasciato il codice sorgente, ha chiuso il suo server e ha fornito finanziamenti per creare The MIT Center for Mobile Learning, guidato dal creatore di App Inventor Hal Abelson e dai colleghi professori del MIT Eric Klopfer e Mitchel Resnick. La versione del MIT è stata lanciata nel marzo 2012. Il 6 dicembre 2013 (l'inizio dell'Ora del Codice), il MIT ha rilasciato App Inventor 2 [10].

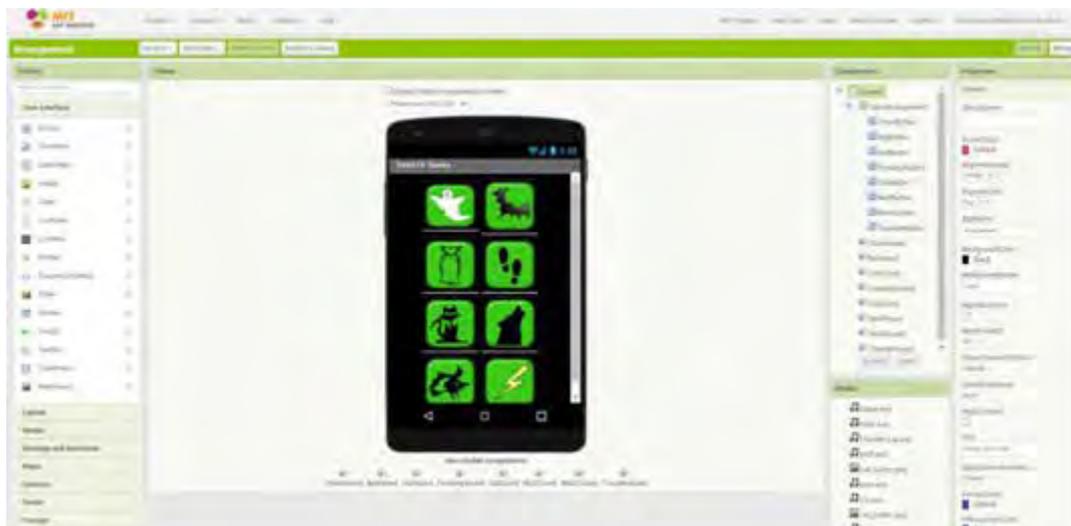


Figura 5
Design editor di App Inventor

L'idea è quella di offrire un luogo in cui gli studenti, a partire dai sei anni, possano apprendere le basi della programmazione con la codifica a blocchi in stile drag-and-drop. È immediatamente e facilmente accessibile: dopo la registrazione, si può iniziare ad utilizzarlo con il solo browser web. MIT App Inventor utilizza blocchi di codice simili a quelli utilizzati dal linguaggio di codifica Scratch.

Per iniziare a sviluppare è indispensabile solamente un browser ed una connessione internet. L'interfaccia utente di MIT App Inventor include due editor principali: il *design editor* ed il *block editor*. Nel *design editor* (fig. 5), mediante drag and drop, si possono disporre gli elementi dell'interfaccia utente (UI) dell'applicazione.

Il *block editor* (fig. 6) è un ambiente in cui gli sviluppatori possono definire visivamente la logica delle loro app utilizzando blocchi codificati a colori che si incastrano insieme come pezzi di un puzzle per descrivere il programma.



Figura 6
Block editor di App Inventor

Durante lo sviluppo si può testare l'applicazione in due modi:

- utilizzando l'emulatore, possibilità che ha il vantaggio di non richiedere neppure uno smartphone ma ha delle limitazioni, in primis quella ovvia di non poterne utilizzare i sensori;
- installando sul proprio smartphone l'app *AI Companion* (disponibile per Android e iOS), si scarica il codice dal pc allo smartphone, tramite cavo usb, rete dati o WiFi, leggendo un QR code all'interno di *Companion*.

Una volta completato lo sviluppo, si può creare l'app in formato APK che si può scaricare ed installare sul proprio smartphone Android. Attualmente non è possibile generare codice che possa essere eseguito in modo autonomo su iOS.

3.2 L'app PythagorHub dal punto di vista dell'utente

L'app ha un'interfaccia molto semplice e di facile utilizzo. Dopo una schermata introduttiva (splash screen) c'è un menù principale dove Archimede Pitagorico ci guida a scegliere fra quattro diverse categorie di dimostrazioni, oppure a sceglierne una casualmente (fig. 7a).



Figura 7°

Il menù principale di PythagorHub

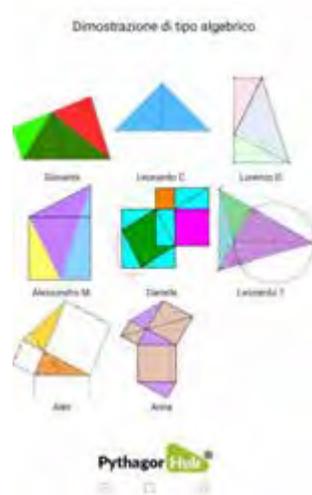


Figura 7b

Uno dei sottomenù

Una volta scelta una categoria, si accede ad un'altra schermata dove si può scegliere fra le specifiche dimostrazioni realizzate dagli studenti (fig. 7b).

Scelta la dimostrazione si apre la schermata con il video della dimostrazione (fig. 8a) e, a richiesta, si può vedere la dimostrazione in forma testuale (fig. 8b).



Figura 8°

Video della dimostrazione

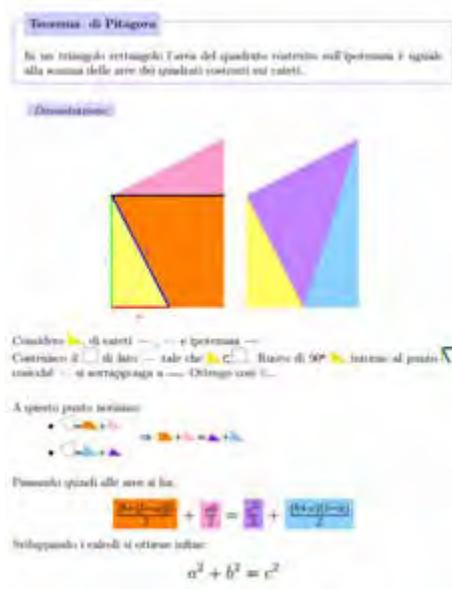


Figura 8b

Versione testuale della dimostrazione

4 Sviluppo dell'app

Gli oggetti (pulsanti, immagini, ...) vanno posizionati su uno schermo (*screen*) nel *design editor* e per ciascun oggetto bisogna poi codificare, nella finestra *block editor*, il comportamento di questi oggetti con costrutti del tipo “se si è verificato questo evento, allora esegui questa azione”. In Figura 9 una implementazione del classico primo programma “hello world” in versione audio.



Figura 9
“Hello World” in versione audio

Per avere un codice più pulito e funzionale, si è deciso di organizzare l'applicazione in più *screen*, otto per l'esattezza:

- uno splash screen;
- il menù principale;
- i 4 screen dove si scelgono le dimostrazioni suddivise in tipologie;
- uno screen dove visualizzare il video e/o il testo della dimostrazione;
- uno screen “about” per scrivere le info minime sull'app.

Un problema da superare è stato quello dovuto al fatto che le variabili, anche quelle globali, sono tali solo relativamente ad uno screen. La scelta effettuata nello screen dove si sceglie la dimostrazione, viene memorizzata in una variabile che dovrebbe essere passata all'interno di quello in cui si visualizza il corrispondente video. Il problema della invisibilità delle variabili fra screen diversi è stato aggirato memorizzandone il valore in uno piccolo database, oggetto standard di App Inventor (letteralmente l'oggetto si chiama *tinyDB*), accessibile da tutti gli screen (fig. 10a e 10b).



Figura 10a
Scrittura nel DB della scelta fatta
nello screen menù



Figura 10b
Lettura nello screen destinato alla riproduzione
del video

Dato che tutti i testi e soprattutto le animazioni video delle dimostrazioni sono incorporate nell'app che non può superare approssimativamente i 50 MB, i video

sono stati drasticamente ridotti di dimensione rispetto a quelli originariamente prodotti. Dato che tecnicamente non sono video complessi e non hanno animazioni rapide, è stato possibile ridurre tutti i video a meno di 1 MB. Non è stato necessario quindi ricorrere a macchinosi workaround che sono reperibili in rete. L'app si installa e si carica rapidamente.

I testi delle dimostrazioni sono stati realizzati in LaTeX. Non dovendo gestire situazioni tipograficamente complesse, una volta fornito un template ed un minimo di istruzioni, i ragazzi sono stati in grado di muoversi in autonomia.

Le figure da inserire nel testo della dimostrazione sono state realizzate con GeoGebra e poi ritagliate. Per inserire le immagini correttamente allineate all'interno del testo non è stato necessario utilizzare un pacchetto specifico, ma si è fatto ricorso semplicemente all'uso della seguente macro:

```
\newcommand*{\img}[1]{%  
  \raisebox{0.0\baselineskip}{%  
    \includegraphics[  
      height=\baselineskip,  
      width=\baselineskip,  
      keepaspectratio,  
    ]{#1}%  
  }%  
}
```

Caricata l'immagine, basta inserire nel testo `\img{nome_immagine}` per avere il risultato voluto.

5 Valutazione dell'esperienza

Il progetto in sé, con la produzione dei contenuti e lo sviluppo dell'app, ha offerto agli studenti coinvolti un nuovo modo di apprendere ed ha costituito per loro un tipo di attività stimolante e rilevante, collegando l'apprendimento ad un oggetto che ha un ruolo centrale nella loro quotidianità.

L'approccio alla geometria utilizzato in questa esperienza ha permesso agli studenti di cambiare la modalità che solitamente hanno di fronte alle dimostrazioni: studiare la versione proposta dal libro di testo o dal docente. In questo contesto i ragazzi hanno assunto un ruolo attivo imparando a ricercare, selezionare e rielaborare dimostrazioni, facendole proprie e sviluppando un metodo efficace per restituirle.

6 Conclusioni

Il progetto è stato sviluppato nell'anno scolastico 2021-2022. In prima battuta si era pensato di sviluppare due percorsi distinti: uno di comprensione autonoma di dimostrazioni presentate non necessariamente in modo tradizionale ed uno di avvio al pensiero computazionale mediante lo sviluppo di app Android. In corso d'opera si è convenuto che la fusione dei due percorsi potesse valorizzare al meglio le due esperienze. Partecipazione e motivazione degli studenti sono state

buone, nonostante il progetto sia stato sviluppato nell'ultima parte dell'anno scolastico quando i ragazzi sono generalmente stanchi ed impegnati nelle verifiche finali. Particolarmente apprezzabili competenze ed interessi dimostrati anche da studenti che durante l'anno si mettono meno in evidenza.

Il prodotto finale è stato presentato per la prima volta in occasione del gemellaggio con un altro Liceo Matematico. Si prevede che i materiali prodotti in questo laboratorio vengano inseriti all'interno di una pagina del sito web della scuola e che il lavoro venga presentato all'utenza esterna in occasione della mostra della matematica organizzata dalla scuola nel prossimo Fibonacci day, il 23 novembre 2022. Le competenze acquisite nello sviluppo dell'app saranno spese dallo stesso gruppo classe per presentare altri contenuti di carattere matematico.

Riferimenti

- [1] Teorema di Pitagora: 118 modi diversi per dimostrarlo
<https://www.cut-the-knot.org/pythagoras/>
- [2] Teorema di Pitagora: versione animata della dimostrazione dei pitagorici
<https://www.youtube.com/watch?v=W62IEEKzSg0>
- [3] Teorema di Pitagora: versione animata della dimostrazione di Airy
<https://www.youtube.com/watch?v=WoRXj-3SG1I>
- [4] Teorema di Pitagora: versione animata della dimostrazione a partire dal primo teorema di Euclide https://www.youtube.com/watch?v=v_S4acaGdBw
- [5] Teorema di Pitagora: versione animata della dimostrazione di Perigal
<https://www.GeoGebra.org/m/atn6F6rv>
- [6] Teorema di Pitagora: versione animata della dimostrazione di Liu-Hui
<https://www.matematicanimata.it/2016/10/19/teorema-di-pitagora-dimostrazione-animata-secondo-liu-hui/>
- [7] https://wiki.geogebra.org/en/ExportImage_Command
- [8] Byrne, O. (William Pickering, 1847), *The first six books of the Elements of Euclid*
- [9] <https://appinventor.mit.edu/>
- [10] https://en.wikipedia.org/wiki/App_Inventor_for_Android

Tirocini informatici online in Istat: esperienze di learning-by-doing di coding e di utilizzo di strumenti low-code

Daniela Casale¹, Maria Pia Di Maio¹ and Giulia Vaste¹

¹ISTAT – Istituto Nazionale di Statistica

casale@istat.it

dimaio@istat.it

vaste@istat.it

1

0

SOMMARIO

La pandemia ha dato un enorme impulso alla formazione a distanza, portando ad esplorarne maggiormente le potenzialità e a sfruttarne i vantaggi. L'Istat propone ogni anno una serie di tirocini formativi, in questo articolo si racconta l'esperienza di due tirocini informatici, svolti in modalità totalmente online, utilizzando strumenti di videoconferenza, di collaborazione, di condivisione di documentazione, di comunicazione sincrona e asincrona. Le tecnologie non solo hanno reso possibile lo svolgimento di incontri frequenti malgrado le distanze geografiche, ma hanno consentito di affiancare gli studenti nel corso dell'apprendimento, guidandoli e supportandoli real-time. Questo aspetto è particolarmente efficace per la natura tecnica dei tirocini proposti, basati sullo studio e l'utilizzo di linguaggi di programmazione e di applicativi low-code. I tirocini sono stati fortemente esperienziali: gli studenti hanno avuto modo di immergersi in un contesto lavorativo reale, collaborando, quasi come componenti del gruppo di lavoro, ad alcune attività di un progetto fondamentale per l'Istituto, la costruzione del Sistema Integrato dei Registri statistici. E' stato possibile personalizzare i contenuti, i feedback, le modalità di svolgimento dei tirocini sulla base delle conoscenze pregresse, degli interessi, delle preferenze, dei tempi di apprendimento e dell'andamento in itinere degli studenti. Ciò ha reso il percorso per i tirocinanti soddisfacente, focalizzato agli obiettivi, efficace, coinvolgente, stimolante, volto ad approfondire le tematiche di maggiore interesse e ad acquisire competenze tecnologiche e comportamentali spendibili nel mondo lavorativo. E' stato dato ampio spazio all'iniziativa individuale, stimolando gli studenti nel proporre soluzioni, accogliendole e valutandole insieme; il

1

0

meccanismo per prove ed errori, caratteristico dello sviluppo di codice e di utilizzo di applicativi, ha consentito di esporre gli studenti a numerose attività che hanno stimolato la capacità di problem solving e la creatività. I tirocini hanno portato a risultati molto soddisfacenti e hanno consentito di realizzare soluzioni efficaci che saranno integrate nei processi produttivi dell'Istituto.

1 Introduzione

La pandemia ha posto l'umanità di fronte a varie sfide, tra cui quella di adattarsi e trovare nuove soluzioni per una didattica da remoto. Le difficoltà riscontrate in alcuni contesti formativi sono indubbie, ma altrettanto evidenti sono le potenzialità e le opportunità che si sono aperte e che è stato possibile cogliere e apprendere per il futuro.

L'Istituto Nazionale di Statistica propone ogni anno una serie di tirocini formativi per studenti universitari, anche volti alla stesura della tesi e al conseguimento della laurea. La pandemia ha favorito la proposta e lo svolgimento di tirocini totalmente online, consentendo facilmente a studenti dislocati in varie parti d'Italia di effettuare tirocini con professionisti delle sedi centrali dell'Istituto. In ambito informatico e tecnologico questa opportunità è particolarmente valida ed efficace poiché non solo lo studente ha modo di fare un'esperienza sul campo relativamente all'organizzazione del lavoro in team da remoto, attraverso l'impiego di tecnologie per la condivisione di materiale, la collaborazione e lo svolgimento di riunioni, ma le tecnologie di fatto lo supportano anche nell'apprendimento, di linguaggi e tecniche di programmazione nonché di software, poiché è possibile guidarlo real-time, seguirlo man mano, intervenendo ove necessario. Il tirocinio diviene esperienziale a tutti gli effetti: lo studente si trova immerso in un contesto lavorativo reale, di un ente di ricerca di rilievo, collabora in progetti in corso cui ha modo di dare il suo contributo, apprendendo da casa come se si trovasse in ufficio, seguito sia nei risultati raggiunti con studio individuale, sia dal vivo affiancato dai tutor in un approccio learning-by-doing (Dewey, 1938). L'esperienza si può classificare in un "e-learning attivo" in cui vengono svolte attività di problem solving, che stimolano il pensiero critico, la creatività nell'individuazione di soluzioni, la rielaborazione delle informazioni fornite (Ranieri, 2005). Le attività sono guidate in itinere da tutor, attraverso l'uso di ambienti di comunicazione asincrona e sincrona per condividere riflessioni, materiali e soluzioni.

Questo articolo illustra lo svolgimento di due tirocini, svolti presso la Direzione Centrale per le Tecnologie Informatiche, in particolare con tutor afferenti al servizio ITE - Servizio Sviluppo e gestione tecnologie a supporto dei registri e delle basi dati, da studenti provenienti dall'Università della Calabria, Corso di Laurea in Statistica ed Informatica per le decisioni e le analisi di mercato. Entrambi i tirocini sono stati svolti in modalità totalmente online, ciò ha consentito di organizzare incontri frequenti, malgrado la distanza geografica tra i partecipanti e gli impegni lavorativi delle tutor, e ha agevolato sia la collaborazione tra gli studenti in una prima fase del tirocinio sia la personalizzazione dei tirocini stessi

sulla base delle conoscenze pregresse, degli interessi, delle preferenze, dei tempi di apprendimento e dell'andamento in itinere degli studenti. In particolare è stato possibile affiancarli e guidarli in attività di analisi e sviluppo di codice funzionante, progettazione di basi di dati, studio di linguaggi e strumenti per la modellazione di processi, utilizzo di software low-code per la realizzazione di applicazioni web (Gartner, s.d.). I tirocini hanno portato a risultati molto soddisfacenti e hanno consentito di realizzare soluzioni efficaci che saranno integrate nei processi produttivi oggetto del tirocinio.

Il paragrafo 2 illustra la prima fase del tirocinio, in cui gli studenti sono stati introdotti al contesto in cui avrebbero lavorato ed è stato definito il percorso di tirocinio personalizzato in base alle conoscenze e interessi di ognuno; il paragrafo 3 descrive la seconda fase più operativa, di svolgimento dei tirocini, con le attività effettuate e i risultati raggiunti; il paragrafo 4 illustra gli strumenti utilizzati, focalizzandosi su cosa è stato possibile fare grazie all'introduzione della tecnologia per questo tipo di esperienza formativa, infine il paragrafo 5 descrive delle brevi conclusioni e sviluppi futuri.

2 Introduzione al contesto e personalizzazione dei percorsi di tirocinio in base a conoscenze e interessi degli studenti

L'Istat negli ultimi anni sta lavorando a un progetto ambizioso e strategico: la costruzione del Sistema Integrato dei Registri statistici (SIR) (Radini, Di Zio, & Vaste, 2021). L'idea base è quella di passare da un approccio «survey based – register assisted» ad uno «register based – survey assisted», che pone i registri, alimentati dalle fonti amministrative, al centro della produzione statistica. La costruzione di un registro è un'operazione molto complessa: si esaminano le fonti amministrative disponibili, si integrano i dati, deduplicandoli, codificandoli ed effettuando delle operazioni di controllo e correzione, si producono degli output annuali e longitudinali nel tempo, che coinvolgono numerose variabili. Tali processi devono essere quanto più possibile monitorati, controllati, supportati anche attraverso strumenti informatici dedicati. Ciò, da un punto di vista tecnico, si traduce nell'individuazione di strumenti opportuni per il lancio e il monitoraggio dei processi e per la produzione e visualizzazione di reportistica. Le tecnologie utilizzate vanno dai linguaggi PL/SQL e SAS per l'implementazione delle procedure di caricamento e manipolazione dei dati, a interfacce web-based che consentano di visualizzare in modo intuitivo l'andamento dei processi. Su questo fronte al momento si stanno utilizzando Apex (Oracle, s.d.), strumento low-code per la realizzazione di applicazioni web prodotto dalla Oracle, e Flows for Apex (Flows for Apex, s.d.), che consente la modellazione di processi secondo lo standard BPMN (Business Process Model and Notation), l'implementazione attraverso l'associazione con codice eseguibile, il lancio e il monitoraggio dei processi attraverso un'interfaccia user-friendly che modifica i colori dei task man mano che vengono eseguiti. Le attività in corso per la costruzione del SIR sono molteplici e includono anche l'introduzione di indicatori per valutare la qualità dei processi. Per i tirocini ci si è focalizzati su alcuni sviluppi per il Registro Tematico

dei Redditi, sull'analisi e progettazione per la definizione di una base dati per l'introduzione di indicatori di qualità nei processi di costruzione dei registri, sull'utilizzo di strumenti per il controllo e il monitoraggio di tali processi, sulla produzione di reportistica.

Una prima fase dei tirocini è stata quindi volta ad introdurre gli studenti alle tematiche relative ai registri: è stata fornita una documentazione generale sul SIR e su cosa si intende per controllo di processo in tale ambito (Vaste, Di Maio, Giorgetti, Passacantilli, & Dell'Orco, 2021), è stata fatta una panoramica sulle attività proposte e alcuni incontri sono stati dedicati a un'analisi delle conoscenze pregresse degli studenti, a un'indagine sui loro interessi e preferenze personali, svolta anche con la collaborazione dei tutor accademici.

In questa prima fase i tirocinanti hanno collaborato tra loro per lo studio delle tematiche e hanno fornito input per definire insieme ai tutor Istat e ai tutor accademici i contenuti e i percorsi più congeniali per i loro tirocini. E' emerso pertanto che in un caso erano già state acquisite competenze sul linguaggio SAS e che quindi era possibile orientarsi su un tirocinio che partisse dall'analisi di codice SAS già sviluppato in Istat per poi tradurlo in procedure PL/SQL e integrarlo nel controllo di processo; nell'altro era forte la propensione per un tirocinio più sperimentale e progettuale per cui ci si è concentrati sulla realizzazione ex novo della base dati a supporto degli indicatori di qualità. Il dettaglio delle esperienze e delle modalità dei tirocini è illustrato nel paragrafo seguente, qui si ritiene importante sottolineare come sia stato possibile personalizzare sia i contenuti dei tirocini sia le modalità di svolgimento in base alle differenze tra i due tirocinanti. Si è coniugata una fase di personale analisi del materiale fornito e di riflessione sulle tematiche proposte, con un'interazione efficace e proficua coi tutor al fine di definire un percorso per entrambi gli studenti soddisfacente, focalizzato agli obiettivi, efficace, stimolante, che consentisse loro di approfondire le tematiche di maggiore interesse e di acquisire competenze tecnologiche spendibili nel mondo lavorativo.

3 Lo svolgimento dei tirocini

I due percorsi di tirocinio hanno avuto un'impronta molto pratica, alternando attività di studio, progettazione, sperimentazione, implementazione individuale a incontri per illustrare e utilizzare insieme gli strumenti o per supportare le attività di progettazione e scrittura del codice. La cadenza degli incontri, nonché le attività richieste sono state costantemente adeguate ai tempi di apprendimento dei tirocinanti. E' stato dato inoltre ampio spazio all'iniziativa individuale, stimolando i tirocinanti nel proporre soluzioni, accogliendole e valutandole insieme. La modalità di apprendimento attraverso l'inserimento in progetti concreti e l'affiancamento di professionisti ha reso i tirocini coinvolgenti per gli studenti, che hanno di fatto lavorato come componenti del gruppo di lavoro. Il meccanismo per prove ed errori, caratteristico dello sviluppo di codice e di utilizzo di applicativi, ha consentito di esporre gli studenti a numerose attività che

hanno stimolato la capacità di problem solving e la creatività, nonché di fornire agli studenti feedback immediati, sia derivanti dai risultati ottenuti, sia dettati dall'esperienza dei tutor. Il primo percorso di tirocinio si è innestato in prodotti già realizzati in Istituto, il secondo è stato più innovativo, definendo la base dati e le prime implementazioni per l'inserimento degli indicatori di qualità nei processi di costruzione dei registri. In entrambi i tirocini sono stati impiegati Apex e Flows for Apex. Un'attività preliminare è stata la predisposizione degli ambienti di sviluppo: i tirocinanti hanno richiesto una workspace che Oracle mette a disposizione gratuitamente per lo sviluppo PL/Sql e l'utilizzo di Apex che è stata popolata con dati di prova forniti dai tutor. In seguito hanno installato l'applicazione Flows for Apex. In tal modo non è stato necessario per i tirocinanti acquistare alcun software dato che hanno potuto scaricare liberamente dai siti di riferimento quanto necessario: questo aspetto può essere molto importante per garantire a tutti gli studenti di poter partecipare ai tirocini informatici proposti da ISTAT.

I tirocinanti sono stati guidati nell'apprendimento e nell'utilizzo di questi software per i casi in esame.

3.1 Il primo percorso di tirocinio

Il primo percorso di tirocinio si è concentrato su due tematiche: da un lato la manipolazione dei dati e dall'altro lo sviluppo di un'applicazione web, analizzando due versioni successive di Flows for Apex ed effettuandone uno studio comparativo.

Il percorso di tirocinio è partito dall'analisi e comprensione di una procedura scritta in linguaggio SAS, relativa al Registro Tematico dei Redditi (RTR), che è stata tradotta in PL/SQL. Il tirocinio ha quindi sollecitato competenze di analisi del codice e dell'attuale base dati da una parte e di sviluppo dall'altra, entrambe con l'affiancamento e il supporto dei tutor. In particolare sono stati suggeriti gli aspetti del PL/Sql su cui focalizzare lo studio fornendo indicazioni volta per volta.

Nell'ambito del tirocinio è stato poi disegnato in BPMN il processo relativo al caricamento del registro, utilizzando Flows for Apex e valutando le differenti funzionalità offerte dalla versione 21.2 rispetto a quella utilizzata attualmente in Istituto. La Figura 1 mostra uno dei flussi realizzati.

Infine è stato affrontato lo studio delle funzionalità di Apex per la realizzazione di Report. Nella Figura 2 se ne riporta un esempio.

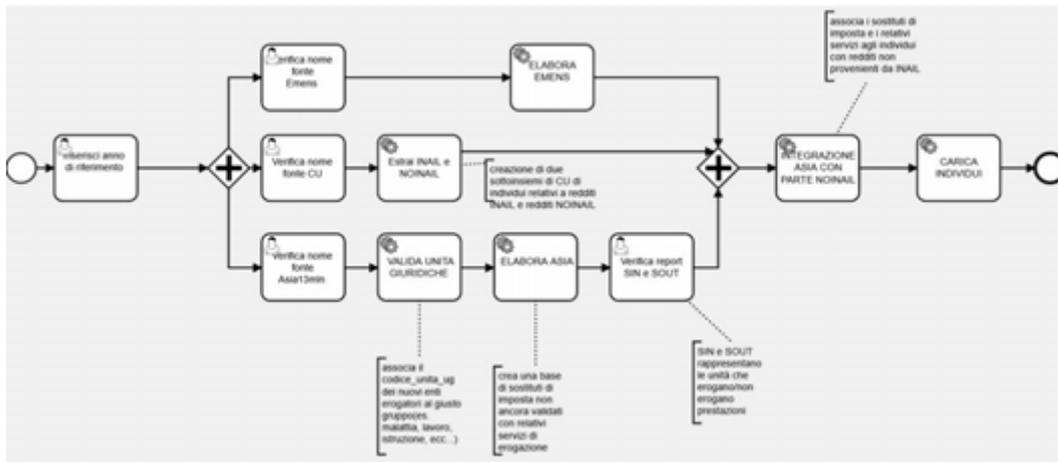


Figura 1

Modellazione di una parte del flusso di RTR in Flows for Apex

	Codice Unita Sig	Regione Sociale	Nome Flag	CA 14	CA 15	CA 17	CA 18	CA 21	Anno Validazione
1	000017	UNIVERSITA' SA...	ITALIA/STRI2	0	0	0	0	1	2015
2	011948	PARIFONDO RL...	ITALIA/ITAL21	1	0	0	0	0	2015
3	019224	ARANDA DI TUT...	ITALIA/MILAN1	1	0	0	0	0	2015
4	020221	UNIVERSITA' VE...	ITALIA/STRI2	0	0	0	0	1	2015
5	021011	REGIONE ABRO...	ITALIA/ENG1	0	0	0	0	0	2015
6	11000005	FONDAZIONE C...	ITALIA/PRE30	0	1	0	0	0	2016
7	0001047	ISTITUZIONE SC...	ITALIA/STRI2	1	0	0	0	1	2016
8	11000106	CASSA RIFORMA...	ITALIA/MILAN1	1	0	0	0	0	2016
9	01100006	ISTITUTO BIRIO...	ITALIA/MILAN1	1	0	0	0	0	2016
10	01040025	ISTITUZIONE SC...	ITALIA/STRI2	0	0	0	0	1	2016
11	010100	UNIVERSITA' EC...	ITALIA/STRI2	0	0	0	0	1	2016
12	10010001	ISTITUZIONE SC...	ITALIA/STRI2	0	0	0	0	1	2016
13	01010111	UNIVERSITA' SO...	ITALIA/MILAN1	0	0	0	0	0	2016

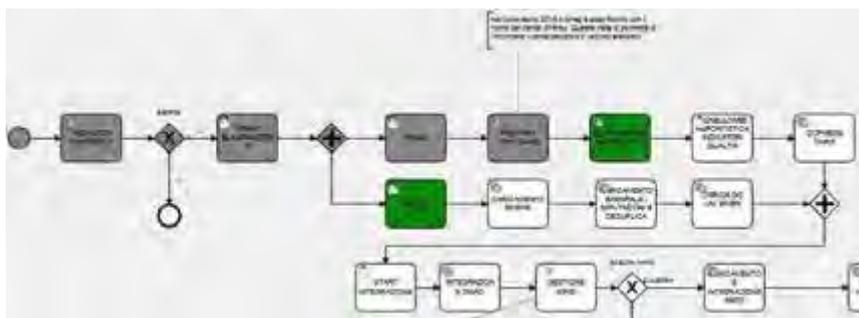
Figura 2

Report unità validate

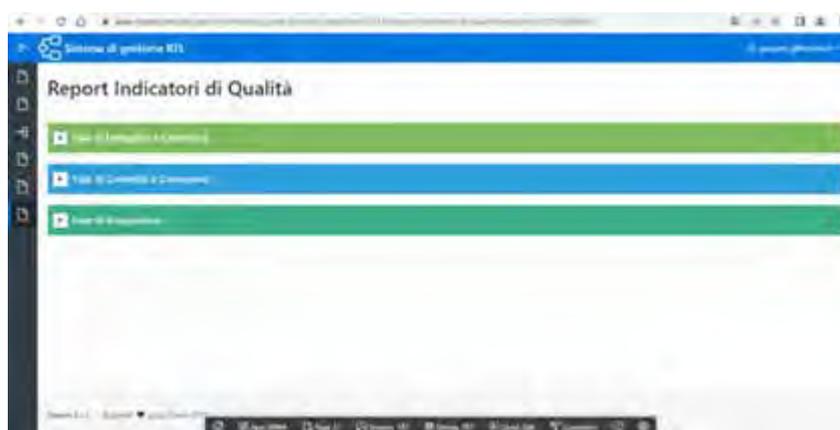
3.2 Il secondo percorso di tirocinio

Il secondo tirocinio si è svolto nell'ambito dell'introduzione di indicatori di qualità nei processi di caricamento dei registri. Il tirocinio ha visto quindi una prima fase di studio di documentazione interna sugli indicatori di qualità, successivamente, in collaborazione coi tutor, è stata progettata la base dati per la memorizzazione degli indicatori, ed è stato implementato il package per il calcolo. Infine sono stati calcolati dei primi indicatori per il processo di caricamento del Registro Tematico del Lavoro (RTL). Una seconda fase del tirocinio si è incentrata sull'inserimento del calcolo degli indicatori nel processo modellato in Flows for Apex e nella

produzione della relativa reportistica in Apex. In Figura 3 viene mostrato un esempio di flusso in esecuzione, in cui sono evidenziati in grigio i task già eseguiti, in bianco quelli ancora da eseguire, in verde i task in esecuzione. Il tirocinante ha quindi sviluppato competenze nell'utilizzo di Apex e di Flows for Apex, ha applicato le conoscenze pregresse per la progettazione della base dati e ha implementato le query e le procedure PL/SQL necessarie per il calcolo degli indicatori. In Figura 4 è riportata la pagina, creata in Apex e integrata nell'applicazione relativa al Registro Tematico del Lavoro, sviluppata per la visualizzazione della reportistica: sono presenti tre sezioni, una per ogni fase del processo per la quale sono stati calcolati degli indicatori (Deduplica e Copertura, Controllo e Correzione, Integrazione). La Figura 5 mostra la sezione sulla fase di Integrazione con una visualizzazione in forma tabellare dell'indicatore "Percentuale di unità derivanti dai differenti dataset" in cui sono evidenziati dei record che violano una condizione predefinita. La Figura 6 mostra i dati della figura precedente in un grafico a torta, che evidenzia in modo intuitivo i contributi dei differenti dataset alle unità finali selezionate nella fase di integrazione.

**Figura 3**

Esecuzione di una parte del flusso di RTL relativa agli indicatori di qualità

**Figura 4**

Pagina di reportistica sugli indicatori di qualità

The screenshot shows a web application interface titled "Report Indicatori di Qualità". It displays a table with several columns: "Indicatore", "Anno", "Data", "Destinatario", "Totale", and "Valore". The table contains multiple rows of data, with some rows highlighted in red to indicate specific values or trends.

Indicatore	Anno	Data	Destinatario	Totale	Valore
4.1 Percentuale di unità derivanti dai differenti dataset sul totale della unità	2020	2020	1	100000000	100000000
4.2 Percentuale di unità derivanti dai differenti dataset sul totale della unità	2021	2021	1	100000000	100000000
4.3 Percentuale di unità derivanti dai differenti dataset sul totale della unità	2022	2022	1	100000000	100000000
4.4 Percentuale di unità derivanti dai differenti dataset sul totale della unità	2023	2023	1	100000000	100000000
4.5 Percentuale di unità derivanti dai differenti dataset sul totale della unità	2024	2024	1	100000000	100000000
4.6 Percentuale di unità derivanti dai differenti dataset sul totale della unità	2025	2025	1	100000000	100000000
4.7 Percentuale di unità derivanti dai differenti dataset sul totale della unità	2026	2026	1	100000000	100000000
4.8 Percentuale di unità derivanti dai differenti dataset sul totale della unità	2027	2027	1	100000000	100000000
4.9 Percentuale di unità derivanti dai differenti dataset sul totale della unità	2028	2028	1	100000000	100000000
4.10 Percentuale di unità derivanti dai differenti dataset sul totale della unità	2029	2029	1	100000000	100000000

Figura 5
L'indicatore "Percentuale di unità derivanti dai differenti dataset" in forma tabellare

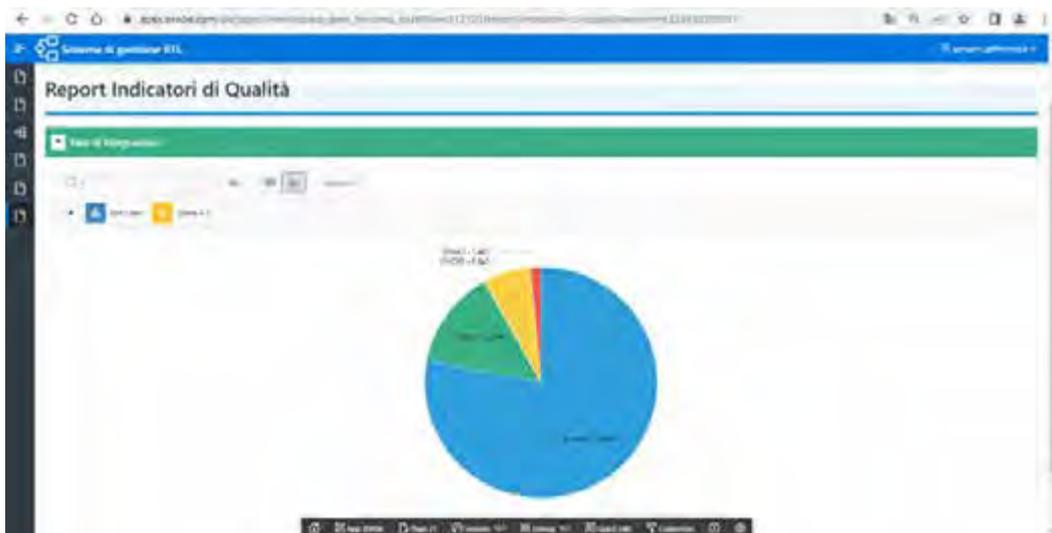


Figura 6
L'indicatore "Percentuale di unità derivanti dai differenti dataset" visualizzato con un grafico a torta

4 Gli strumenti utilizzati

Le riunioni sono state svolte utilizzando lo strumento per videoconferenze in uso in Istituto. In particolare per le attività di coding e di guida all'utilizzo di Apex e Flows for Apex è stata ampiamente utilizzata la funzionalità di "condivisione dello schermo", che ha consentito un affiancamento real-time, guidando lo studente non solo nello sviluppo, ma anche nel test e nel debug di quanto realizzato.

In particolare, lo studio comparativo tra la versione 21.2 di Flows for Apex e la versione attualmente in uso in Istituto è stato affrontato sia con attività asincrone di riflessione individuale, sia con incontri mirati di condivisione e confronto. Un'attività di questo tipo ha sollecitato lo spirito critico e la capacità di analisi degli studenti, impegnandoli in una valutazione dei pro e dei contro degli strumenti, in termini di interfacce utente e funzionalità disponibili. Un compito specifico è stato lo studio di come inserire i parametri di input da interfaccia e durante l'esecuzione del flusso, con l'obiettivo di individuare la soluzione più user-friendly, più intuitiva e semplice da usare per l'utente finale. Ciò ha sollecitato le capacità di analisi, di problem solving e lo spirito critico degli studenti per individuare le possibili soluzioni implementative e selezionare la più opportuna. Tali capacità sono state sollecitate anche per le attività di progettazione della base dati per gli indicatori di qualità: a partire dallo studio teorico di tali indicatori e dall'osservazione concreta di un caso d'uso sul registro tematico del lavoro, il tirocinante ha applicato le sue conoscenze pregresse sui database e, coadiuvato in riunioni online dedicate dalle tutor, ha predisposto lo schema concettuale, logico e fisico delle tabelle necessarie. Inoltre ha analizzato il processo di caricamento del registro e ha individuato dove calcolare gli indicatori, applicando e implementando le regole di calcolo, dimostrando capacità di analisi, spirito propositivo, competenze proprie di un analista sviluppatore software. La reportistica prodotta è stata progettata insieme, stimolando la creatività dello studente, ad esempio nel rendere immediatamente individuabili visivamente i valori anomali. I tirocinanti sono stati molto attivi, hanno svolto le attività proposte in modo proattivo, con suggerimenti e proposte di soluzione. Le attività sono state costantemente supportate dalle tutor, in modo asincrono via mail, ma soprattutto attraverso l'utilizzo della videoconferenza che ha consentito un affiancamento real-time, grazie al quale lo studente è stato sostenuto e indirizzato, attraverso l'illustrazione di esempi, soprattutto per la reportistica, il supporto alla comprensione, specie per imparare ad utilizzare gli strumenti Apex e Flows for Apex o per interpretare il codice SAS e utilizzare il linguaggio PL/SQL e la notazione BPMN, l'integrazione di informazioni di contesto, laddove la documentazione fornita non era risultata sufficiente.

Per fornire un'assistenza più valida, sono stati inviati al momento opportuno ai tirocinanti tutti i link utili per trovare esempi o casi di studio realizzati con gli strumenti software impiegati nei tirocini. Sono stati, inoltre, consegnati loro dei template e degli esempi sviluppati in Istituto da usare come base di partenza per lo sviluppo. Questa modalità è stata molto aiutata dall'impiego di strumenti low code che ha consentito di concentrarsi sulle finalità, strutturazione e

documentazione del codice piuttosto che sulla difficoltà delle istruzioni di programmazione.

Questa metodologia di svolgimento dei tirocini è risultata molto efficace.

Anche per la documentazione di quanto realizzato, al fine della scrittura della tesi, gli strumenti di collaborazione, condivisione di documentazione, comunicazione sincrona e asincrona sono stati preziosi, consentendo una revisione agile e condivisa, sostituendo pienamente i supporti cartacei.

5 Conclusioni e sviluppi futuri

L'utilizzo delle tecnologie ha in primis consentito un semplice ed efficace svolgimento dei tirocini descritti in questo articolo: pre-pandemia sarebbe stato difficile immaginare di condurli non in presenza, esclusivamente con riunioni online. Inoltre è stato possibile seguire gli studenti e personalizzare le attività, le modalità, i feedback, i tempi dell'apprendimento in base alle loro esigenze e preferenze, affiancandoli in attività svolte insieme online. I risultati sono stati molto soddisfacenti e sono state realizzate soluzioni efficaci che saranno integrate nei processi produttivi dell'Istituto. L'esperienza è stata assolutamente positiva e apre alla possibilità di tirocini svolti in collaborazione tra studenti dislocati ovunque sul territorio, favorendo lo sviluppo di soft skills quali la capacità di lavorare in team eterogenei, la focalizzazione sugli obiettivi, il problem solving, la comunicazione.

Bibliografia

- Dewey, J. (1938). *Experience and education*.
Flows for Apex. (n.d.). *Flows for Apex*. Retrieved from <https://flowsforapex.org/>
Gartner. (n.d.). *Enterprise Low-Code Application Platforms (LCAP) Reviews and Ratings*. Retrieved from Gartner Peer Insights: <https://www.gartner.com/reviews/market/enterprise-low-code-application-platform>
Oracle. (n.d.). *Oracle Apex*. Retrieved from <https://apex.oracle.com/it/>
Radini, R., Di Zio, M., & Vaste, G. (2021). Il Sistema Integrato dei Registri dell'Istat: un sistema di dati integrato a servizio del Paese. *Presentazione alla XIV Conferenza Nazionale di Statistica 30 nov-1 dic 2021*. Roma.
Ranieri, M. (2005). *E-learning: modelli e strategie didattiche*. Erickson.
Vaste, G., Di Maio, M. P., Giorgetti, P., Passacantilli, M., & Dell'Orco, F. (2021). Il sistema integrato dei registri (SIR): sperimentazione su strumenti di controllo di processo. *Poster in XIV Conferenza Nazionale di Statistica 30 nov-1 dic 2021*.