



UNIVERSITÀ DEGLI STUDI DI PADOVA

Dipartimento di Psicologia dello Sviluppo e della Socializzazione

**Corso di Laurea Triennale in Scienze Psicologiche dello Sviluppo, della Personalità e delle
Relazioni Interpersonali**

Tesi di Laurea Triennale

**Differenze di genere nel pensiero computazionale: apprendimento
e percezione di autoefficacia prima e dopo un intervento di coding
in bambine e bambini della scuola primaria**

**Gender differences in computational thinking: learning and perceived self-efficacy
before and after a coding intervention in girls and boys in primary school**

Relatrice

Prof.ssa Barbara Arfè

Correlatrice

Dott.ssa Costanza Padova

Laureanda: Sacchetto Martina

Matricola: 2054024

Anno Accademico 2023/2024

INDICE

INTRODUZIONE	1
CAPITOLO I: PENSIERO COMPUTAZIONALE	3
1.1 Che cos'è il pensiero computazionale?	3
1.2 Caratteristiche peculiari del pensiero computazionale	4
1.3 Importanza del pensiero computazionale a scuola	5
1.4 Strumenti per allenare il pensiero computazionale	7
CAPITOLO II: DIFFERENZE DI GENERE	9
2.1 Stem e differenze di genere	9
2.2 Pensiero computazionale e differenze di genere: studi	11
2.3 Differenze di genere nella percezione di autoefficacia nel pensiero computazionale	12
CAPITOLO III: LO STUDIO	15
3.1 Obiettivo di ricerca	15
3.2 Partecipanti	16
3.3 Strumenti.....	16
3.4 Procedura	19
CAPITOLO IV: RISULTATI	24
4.1 Analisi preliminari.....	24
4.2 Differenze di genere nell' autoefficacia al pre-test	26
4.3 Differenze di genere nell'autoefficacia al post-test.....	28
4.4 Differenze di genere nella variazione dell'autoefficacia	31
LIMITI, PROSPETTIVE FUTURE E CONCLUSIONE	36
BIBLIOGRAFIA	38
SITOGRAFIA	44

INTRODUZIONE

Il pensiero computazionale (CT), considerato una competenza fondamentale del XXI secolo, rappresenta la capacità di risolvere problemi, progettare sistemi e comprendere il comportamento umano utilizzando i concetti fondamentali dell'informatica (Wing, 2006). Questa abilità è stata integrata nei curricula scolastici per sviluppare abilità trasversali in vari ambiti, grazie anche all'utilizzo di strumenti come il *coding* (Arfé et al., 2019). Il divario di genere nel CT è documentato a livello globale, con studi che riportano risultati contrastanti: ad esempio, Funke e Geldreich (2017) hanno indagato l'effetto di un corso sulle competenze fondamentali di programmazione, utilizzando Scratch con alunne e alunni di età compresa fra i 5 e i 12 anni. I risultati non hanno evidenziato differenze di genere nelle abilità di programmazione acquisite dai partecipanti. Diversamente, uno studio di Román-González et al. (2017) condotto su bambini spagnoli dalla quinta alla decima elementare ha esaminato la relazione tra le capacità di pensiero computazionale e il genere, rivelando una differenza significativa a favore dei maschi nei punteggi del "*Computational Thinking test*". I risultati indicano che i ragazzi ottengono punteggi mediamente più alti rispetto alle ragazze, con un divario che aumenta con l'avanzare del grado scolastico.

Questo studio si è posto l'obiettivo di esplorare le differenze di genere durante il processo di apprendimento del *coding* e di analizzare sia la percezione di autoefficacia di alunne e di alunni prima e dopo l'intervento, sia come quest'ultima possa variare nel tempo. La letteratura su questo argomento presenta risultati dissimili: ad esempio, Tellhed et al. (2022) hanno riscontrato che i maschi mostrano livelli di autoefficacia più elevati rispetto alle femmine nelle loro capacità di programmazione e nelle discipline STEM mentre Gunbatar e Karalar (2018), nella loro indagine sugli effetti dell'insegnamento della programmazione, utilizzando mBlock, sulle percezioni di autoefficacia e sugli atteggiamenti delle studentesse e degli studenti delle scuole medie, hanno stabilito che sebbene, all'inizio della ricerca, le percezioni di autoefficacia dei ragazzi nei confronti della programmazione fossero superiori a quelle delle ragazze, questa differenza veniva colmata alla fine della ricerca.

Alla ricerca hanno partecipato 89 alunni (M = 40, F = 49) di cinque classi quarte provenienti da tre diverse scuole primarie. I partecipanti sono stati assegnati in modo casuale a due gruppi: 55 bambini al gruppo sperimentale (che hanno partecipato ad un training di *coding* dalla durata di un mese) e 34 bambini al gruppo di controllo (avviati al *coding* successivamente). Il disegno sperimentale prevedeva tre momenti di valutazione: prima dell'intervento (T1), durante il training e dopo l'intervento (T2). A T1 e T2, alle alunne/i sono stati proposti quattro esercizi di difficoltà crescente, ciascuno preceduto da un esercizio simile di esempio e un questionario (ESCAS) per indagare la loro percezione di autoefficacia relativa alla programmazione/*coding* e il loro interesse nei confronti di quest'ultimo.

Gli studenti avevano il compito di creare un programma che guidasse un personaggio attraverso un percorso trascinando dei blocchi che rappresentavano comandi specifici.

I risultati emersi da questa ricerca suggeriscono che il training di *coding* proposto è stato efficace per entrambi i generi, con una differenza marginalmente significativa a favore dei maschi. Per quanto riguarda i livelli di autoefficacia legata alla programmazione, non sono state riscontrate differenze di genere significative né prima né dopo l'intervento. Tuttavia, è rilevante il divario di genere riscontrato nella variazione dell'autoefficacia, che è a favore dei maschi. Al contrario, nella sottoscala *Confidence*, i risultati sembrano essere a favore delle femmine.

Lo studio si articola in due parti: i primi due capitoli, che costituiscono la prima parte, hanno lo scopo di fornire un quadro teorico sul pensiero computazionale, l'autoefficacia e la presenza di differenze di genere in questi ambiti. La seconda parte illustra la ricerca sul campo. Nel primo capitolo viene data una definizione di *Computational Thinking*, sono individuate le sue caratteristiche peculiari e viene ribadita l'importanza di introdurlo nelle scuole fin dalla prima infanzia. Inoltre, è descritto, in base alle evidenze presenti in letteratura, uno strumento, risultato efficace per lo sviluppo delle abilità del pensiero computazionale: *il coding*. Nel secondo capitolo sono trattate sia le differenze di genere in ambito STEM sia gli studi che riscontrano la presenza o meno di questo divario nel pensiero computazionale. In più, è fornita una definizione di autoefficacia, partendo dalle formulazioni teoriche dello psicologo Albert Bandura, descrivendo i fattori che la influenzano, le strategie per incrementarla e le differenze di genere in questo ambito.

Il terzo capitolo, nucleo centrale di questa tesi, analizza e sintetizza lo studio condotto. Viene spiegato il metodo di ricerca, illustrando l'obiettivo e le ipotesi. Sono, quindi, presentati gli strumenti utilizzati per valutare sia il livello di autoefficacia che le abilità di *coding*. Infine, è descritta la procedura di intervento.

Le modalità di analisi dei dati, i risultati in merito alle differenze di genere nell'apprendimento del *coding* e nell'autoefficacia prima e dopo l'intervento, sono argomento del quarto e ultimo paragrafo. Infine, sono rilevati i limiti dello studio ed evidenziati spunti per eventuali possibili sviluppi futuri.

CAPITOLO I

PENSIERO COMPUTAZIONALE

1.1 CHE COS'È IL PENSIERO COMPUTAZIONALE?

L'espressione "*Computational Thinking*" (CT) fu resa nota per la prima volta nel 1980 dal matematico e pedagogista sudafricano Seymour Papert nel suo libro *Mindstorms: Children, Computers and Powerful Ideas* nel quale suggerì che i computer potessero migliorare il pensiero e cambiare i modelli di accessibilità alla conoscenza (Papert, 1980). Ha inoltre sottolineato che tutti i bambini dovrebbero avere accesso a questi dispositivi come un modo per plasmare il loro apprendimento ed esprimere le loro idee (Papert, 1996). Più tardi, nel suo influente articolo, Wing (2006) ha fatto eco a queste idee illuminando il concetto di CT e definendolo come il processo mentale che consente di risolvere problemi, progettare sistemi e comprendere il comportamento umano attingendo ai concetti fondamentali dell'informatica. Wing descrive il pensiero computazionale come un'abilità fondamentale per tutti, non solo per gli informatici. Questa capacità riguarda la risoluzione di problemi, la progettazione e la comprensione del comportamento umano, basandosi sui concetti fondamentali dell'informatica. Nel 2010, Wing ha ulteriormente approfondito la definizione di CT, descrivendolo come l'insieme dei processi di pensiero coinvolti nella formulazione di problemi e delle relative soluzioni, rappresentate in modo tale da poter essere eseguite efficientemente da un agente di elaborazione delle informazioni. Questa definizione venne poi semplificata da Aho (2012), che concettualizzò il CT come i processi di pensiero coinvolti nella formulazione dei problemi in modo che le loro soluzioni possano essere rappresentate sotto forma di passaggi computazionali e algoritmi.

L'associazione degli insegnanti di informatica (CTSA) e l'International Society for Technology in Education (ISTE) hanno fornito una definizione più operativa del *Computational thinking* descrivendolo come un processo di risoluzione dei problemi che include (ma non si limita a) diverse caratteristiche essenziali:

- la capacità di formulare problemi in modo tale da poterli risolvere attraverso l'utilizzo di computer o altri strumenti;
- l'organizzazione e l'analisi logica dei dati;
- la rappresentazione dei dati attraverso astrazioni come modelli e simulazioni;

- l'automatizzazione delle soluzioni tramite il pensiero algoritmico (una serie di passaggi ordinati); l'identificazione, l'analisi e l'implementazione di soluzioni ottimali con l'obiettivo di ottenere la combinazione più efficiente ed efficace di passi e risorse;
- la capacità di generalizzare e applicare questo processo di problem solving a una vasta gamma di problemi.

Secondo una definizione orientata all'ambito educativo il pensiero computazionale può essere visto come un insieme di concetti computazionali (sequenze, loop, parallelismo, eventi, condizionali, operatori e dati), pratiche computazionali (sviluppo incrementale e iterativo, test, debug, riutilizzo, remix, astrazione, modularizzazione) e prospettive computazionali (essere, connettere, interrogarsi) (Brennan & Resnick, 2012).

Le articolate definizioni teoriche di Papert, Wing e Aho del pensiero computazionale sono state, quindi, successivamente declinate in modo più settoriale sia nell'ambito operativo dell'informatica sia nell'ambito educativo e curricolare.

1.2 CARATTERISTICHE PECULIARI DEL PENSIERO COMPUTAZIONALE

Secondo Arfè e Vardanega (2019) “il pensiero computazionale riguarda una forma specifica di ragionamento: il ragionamento attorno ai problemi” (p.766). Questa forma di ragionamento non coinvolge abilità intuitive, ma forme di ragionamento analitico, ossia di pensiero lento. Tuttavia, il pensiero computazionale può includere processi di ragionamento deduttivo, induttivo, abduttivo e analogico. Wing (2006) sostiene che CT non significa pensare come un computer; ma impegnarsi in cinque processi cognitivi con l'obiettivo di risolvere i problemi in modo efficiente e creativo. Questi includono:

1. **decomposizione del problema:** un processo analitico e deduttivo che comporta l'analisi e la scomposizione di un problema complesso in una serie di sotto-problemi per giungere ad una soluzione complessiva.
2. **astrazione:** un processo induttivo che permette la semplificazione e la categorizzazione dei problemi, individuando i dettagli rilevanti da considerare e quelli non necessari.
3. **generalizzazione:** capacità di individuare similitudini o schemi ricorrenti e di trasferire le procedure di soluzione del problema a una varietà di nuove sfide.
4. **pensiero algoritmico:** definire una serie di azioni o istruzioni di base che fungono da direttive per chi esegue l'operazione, con l'obiettivo di raggiungere la soluzione del problema o l'obiettivo desiderato.

5. **verifica logica:** capacità analitica di esaminare attentamente il modello mentale di risoluzione di un problema al fine di individuare e correggere gradualmente gli errori negli algoritmi elaborati (debugging) e di identificare procedure più efficienti e istruzioni più sintetiche.

Nel rapporto completo del National Research Council (NRC, 2010), il pensiero computazionale viene descritto attraverso cinque elementi essenziali e universali applicabili in tutti i domini:

- 1) test di ipotesi: quando si risolve un problema complesso in qualsiasi ambito, è fondamentale generare e testare ipotesi in modo sistematico per comprendere il funzionamento del sistema. Poiché non è possibile testare tutte le variabili, è cruciale selezionare i parametri più pertinenti da verificare;
- 2) gestione dei dati: la gestione dei dati comporta la raccolta di informazioni provenienti da diverse fonti, l'elaborazione di modelli di dati e la rappresentazione dei dati in modo significativo;
- 3) parallelismo: il parallelismo si riferisce all'elaborazione simultanea di informazioni da più fonti o dimensioni;
- 4) astrazione: l'astrazione si concentra sulla modellazione del funzionamento di un problema/sistema complesso;
- 5) debug: si riferisce alla ricerca e alla correzione di errori dopo aver costruito particolari modelli.

1.3 IMPORTANZA DEL PENSIERO COMPUTAZIONALE A SCUOLA

Il pensiero computazionale è stato definito da Curzon et al. (2009) la competenza del XXI secolo. Con il progredire della tecnologia e il coinvolgimento sempre maggiore dei computer e delle soluzioni computazionali nella nostra vita quotidiana, tutti i livelli di istruzione devono orientarsi verso la preparazione delle giovani menti a un mondo sempre più digitale. Secondo Wing (2006), il pensiero computazionale dovrebbe essere appreso e utilizzato da tutti e non unicamente da chi lavora in ambito accademico o più strettamente informatico. L'autrice promuove il CT come un'abilità vitale per il presente e il futuro, equiparandone l'importanza a quella della lettura, della scrittura dell'aritmetica di base. Il computer diventa quindi un mezzo di semplificazione e supporto nella formulazione e nella risoluzione dei problemi.

Negli ultimi anni si è sviluppata la volontà di integrare l'informatica, e quindi il pensiero computazionale, nell'istruzione disciplinare, in particolare nei campi delle scienze, della tecnologia, dell'ingegneria e della matematica (STEM) (Lee et al., 2020; Li et al., 2020). La ricerca ha dimostrato che i contenuti e i contesti STEM possono facilitare l'apprendimento del CT e contribuire a ridurre le disuguaglianze nel suo apprendimento (Orton et al., 2016; Weintrop et al., 2014). Allo stesso modo, l'integrazione del CT nell'istruzione STEM migliorerà anche l'apprendimento dei contenuti STEM da parte delle/degli studenti (Repenning et al., 2010; Sengupta et al., 2013; Wilensky & Reisman, 2006;

Lewis & Shah, 2012), grazie al ruolo centrale del calcolo nelle discipline STEM moderne (Martin, 2018).

Nelle scuole italiane il pensiero computazionale è stato introdotto nell'anno scolastico 2014/2015 con il progetto "Programma il futuro" grazie al quale sono stati forniti alle scuole degli strumenti divertenti e accessibili per formare studentesse e studenti ai concetti di base dell'informatica e del CT. Il progetto anticipa gli obiettivi contemplati dalla legge 107/2015¹, la quale non lo equipara esplicitamente alle altre discipline scolastiche, ma lo include tra gli obiettivi per le competenze digitali. Nel 2018 invece, un Comitato Scientifico Nazionale istituito dal MIUR, oggi MIM, ha redatto un documento denominato «Indicazioni nazionali e nuovi scenari» per il curriculum della scuola dell'infanzia e del primo ciclo di istruzione. Il testo stabiliva che tra gli obiettivi dell'istruzione primaria vi fosse l'introduzione del pensiero computazionale, integrando ufficialmente questa nuova disciplina nel curriculum scolastico obbligatorio. Specificava inoltre che, al pari di altre discipline (matematica, lingue, storia e geografia), il suo fine è educare a una forma di pensiero.

L'introduzione del pensiero computazionale fin dai primi anni di scuola non ha l'obiettivo di insegnare un particolare strumento (linguaggio) di programmazione, ma di insegnare a ragionare, a partire da problemi che prevedono soluzioni computazionali, cioè elaborazioni algoritmiche costituite da istruzioni eseguibili da terzi (una macchina o un agente umano), impiegando per esse le procedure sistematiche dell'informatico (Nardelli & Ventre, 2015; Wing, 2006). L'insegnamento del pensiero computazionale, quindi, non richiede necessariamente la programmazione (*coding*). Le attività, infatti, possono essere sia digitali (usando il computer come esecutore terzo) che analogiche (usando, per esempio, il linguaggio naturale o il disegno per definire una sequenza di istruzioni necessarie per permettere a una terza persona di scoprire una regola o identificare una categoria di oggetti). In entrambi i casi, le attività coinvolte possono essere di due tipi:

- 1) attività di *problem solving*, più o meno strutturate (giochi logici) (Kalelioğlu, 2015), e questo è il caso del ragionamento deduttivo e abduttivo;
- 2) attività di progettazione, volte a sviluppare idee o prodotti (ad esempio, un videogioco, una storia), e tipicamente meno strutturate (Bers, 2018; Strawhacker & Bers, 2019).

L'insegnamento del pensiero computazionale a scuola consente di apprendere ad affrontare le situazioni in modo analitico, scomponendole nei vari aspetti che le caratterizzano e pianificando per

¹ <https://www.gazzettaufficiale.it/eli/id/2015/07/15/15G00122/sg>

ognuno le soluzioni più idonee. Il pensiero computazionale ha il potenziale per offrire agli studenti molto più di una semplice "alfabetizzazione tecnologica," intesa come conoscenza pratica dell'uso del computer nelle attività quotidiane. Esso permette loro di essere più efficaci nella risoluzione di problemi in contesti diversi dall'informatica e li incoraggia a creare nuovi strumenti per affrontare le sfide, anziché limitarsi a utilizzare quelli esistenti (Phillips, 2007).

Nonostante il pensiero computazionale comprenda una serie di principi fondamentali dell'informatica, esso non si limita esclusivamente a questo ambito e ai settori tecnologici (Hunt 2012). Anche se uno studente intraprende una carriera diversa dall'informatica, le competenze acquisite e sviluppate attraverso il CT saranno utili in qualsiasi campo professionale. A scuola, ragionamento analogico, deduttivo e induttivo non sono specificità esclusiva del pensiero computazionale, ma possono caratterizzare varie attività di apprendimento e di pensiero «lento», tra cui comprendere e produrre testi (Klein et al., 2014; Van Den Broek, 1989), comprendere e risolvere problemi matematici (Francisco & Maher, 2005), dibattere di idee o concetti (Mason, 2001).

1.4 STRUMENTI PER ALLENARE IL PENSIERO COMPUTAZIONALE

Negli ultimi anni gli scienziati hanno cercato di individuare strumenti che potessero allenare la capacità di pensiero computazionale a tutti i livelli di istruzione, arrivando ad inserire questa disciplina all'interno dei curriculum scolastici (Tsarava et al., 2022). Il *coding* si è dimostrato essere uno strumento valido e concreto per allenare il pensiero computazionale (Arfè et al., 2019; Flórez et al., 2017; Sáez-López et al., 2016). Consiste nel generare istruzioni (codice del programma) in modo da ottenere piani eseguibili, il cui effetto sul problema può essere accertato empiricamente.

L'insegnamento del *coding* può avvenire sia tramite computer che senza il supporto di strumenti digitali, ad esempio con carta e penna (Arfè et al., 2019). Si possono distinguere:

- *coding plugged*: attività di apprendimento che utilizzano computer, dispositivi elettronici o altri strumenti digitali per migliorare le capacità di pensiero computazionale degli studenti (Grover & Pea, 2013.). Prevede l'utilizzo di piattaforme come Code.org per lo svolgimento di esercizi di *coding* virtuale.
- *coding unplugged*: attività didattiche che insegnano i concetti di base della programmazione senza l'uso di computer o dispositivi elettronici. Queste attività utilizzano materiali semplici come carta, matite, o giochi fisici.

In entrambi i casi, l'obiettivo è quello di insegnare strategie di problem solving, introducendo concetti base come gli operatori logici, le sequenze e il debugging, ovvero l'individuazione degli errori (bug) del programma per poi correggerli (Arf  et al., 2019).

CAPITOLO II

DIFFERENZE DI GENERE

2.1 STEM E DIFFERENZE DI GENERE

Con differenze di genere, o gender gap, nell'ambito delle discipline scientifiche, tecnologiche, ingegneristiche e matematiche (STEM), ci si riferisce alla disparità tra ragazze e ragazzi, donne e uomini, in termini di prestazioni, risultati, interessi o convinzioni in questi settori (Montuori et al., 2022). Nonostante i progressi degli ultimi decenni, le donne continuano ad essere sottorappresentate nei campi STEM, sia nell'istruzione sia nel mercato del lavoro. Questa disparità non solo limita le opportunità per le donne, ma anche il potenziale innovativo e creativo delle discipline stesse. L'inclusione delle donne è importante per promuovere l'eccellenza scientifica e migliorare la qualità dei risultati nelle STEM, poiché le diverse prospettive aumentano la creatività, riducono i pregiudizi e favoriscono conoscenza e soluzioni più solide (UNESCO). L'Unione europea (UE) ha incluso tra le priorità l'aumento della partecipazione femminile alle STEM, sviluppando programmi di sostegno come la Piattaforma europea delle donne scienziate (EPWS) e la Gender Equality Network in the European Research Area (GENERA) per colmare questo divario. La *“Leaky pipeline”*, letteralmente tubo che perde, è la metafora che viene utilizzata oggi per indicare il fenomeno di perdita del capitale umano femminile durante i percorsi di carriera STEM. Il bacino di potenziali lavoratori in questo campo è piuttosto ampio tra le studentesse e gli studenti del primo ciclo di istruzione. Tuttavia, molti di loro perdono interesse man mano che avanzano nei loro studi, scegliendo di concentrarsi su altre discipline: sostanzialmente il grado di interesse per le discipline STEM è inversamente proporzionale al grado di istruzione frequentato. Sebbene, negli ultimi anni, la partecipazione delle donne all'istruzione e alle carriere STEM sia aumentata (Miyake et al., 2010), il divario di genere in queste aree persiste. Solo circa il 30% di tutte le studentesse sceglie di studiare materie legate alle discipline STEM nell'istruzione superiore (ONU, 2019.) e solo il 3% di queste decide di studiare materie legate alla tecnologia (UNESCO, 2017). I dati ISTAT relativi al 2022 riportano che nella fascia di età tra i 25 e i 34 anni, nonostante i livelli di istruzione femminile siano più elevati rispetto a quelli di istruzione maschile (23,5% delle donne, contro il 17,1% degli uomini), tra i laureati in discipline STEM esiste un importante divario di genere. Per gli uomini la percentuale di laureati STEM è del 34,5% (un laureato su tre), mentre per le donne la percentuale scende a 16,6% (una laureata su sei).

I ricercatori offrono due spiegazioni distinte all'emergere delle differenze di genere nelle materie STEM.

Da una parte, sostengono che il divario di genere ha origine da differenze innate nelle abilità cognitive che stanno alla base della performance in questo ambito; tali disparità possono influenzare i risultati delle studentesse e degli studenti (Maloney et al., 2012) e, di conseguenza, la motivazione a perseguire studi e carriere nelle discipline STEM (Wang & Degol, 2017).

Dall'altra ritengono che i fattori socioculturali come le disuguaglianze nei sistemi sociali e educativi e gli stereotipi di ruolo di genere che determinano pregiudizi, sia impliciti che espliciti, condizionano il modo in cui ragazze e ragazzi percepiscono e valutano le proprie e altrui prestazioni e abilità nelle STEM (Charlesworth & Banaji, 2019; Girelli, 2022). Per spiegare come questi stereotipi di genere influenzino le scelte e le percezioni individuali, il concetto di *habitus* del sociologo Pierre Bourdieu (1990) è particolarmente illuminante. L'autore con il seguente termine descrive l'insieme di predisposizioni e schemi di pensiero che le persone sviluppano attraverso le esperienze sociali. L'*habitus* è il meccanismo che permette agli individui di prendere decisioni, intraprendere scelte, osservare la realtà, saperla analizzare per darle un significato. Ogni azione, che sia volontaria o meno, deriva da un'elaborazione implicita dell'*habitus*. Quello che avviene è che gli individui, interiorizzando le norme e le aspettative di genere della società, sviluppano disposizioni che influenzano le loro aspirazioni, le loro scelte formative e professionali e il modo in cui interpretano il proprio successo o fallimento. Questi stereotipi influenzano le persone fin dalla prima infanzia e possono determinare le loro scelte di carriera. Infatti, attualmente, sono poche le donne che decidono di iscriversi a corsi di ingegneria, mentre sono molte quelle che prediligono corsi universitari di tipo umanistico. L'idea è che i condizionamenti sociali, come quelli ricevuti durante l'infanzia attraverso la socializzazione, vengano interiorizzati dall'*habitus*, in questo contesto il genere: fattore che ci induce a scartare scelte considerate "troppo inusuali". Master et al. (2017) hanno osservato che già a 6 anni i bambini sviluppano stereotipi di genere, credendo che i ragazzi dovrebbero essere più interessati e più bravi nella programmazione e nella robotica rispetto alle ragazze. Per contrastare questa disparità, è importante suscitare l'interesse delle ragazze per le STEM fin da piccole, prima che gli stereotipi siano profondamente radicati (Metz, 2007; Steele, 1997). La ricerca mostra che i bambini coinvolti in attività STEM durante l'infanzia sviluppano meno stereotipi di genere e incontrano meno ostacoli nel perseguire carriere in questi campi (Metz, 2007). Un risultato significativo per sostenere l'idea che i condizionamenti relazionali hanno un ruolo decisivo nel determinare il divario nelle STEM, è quello emerso nello studio di Eisenkopf et al. (2015) in cui i ricercatori hanno voluto esaminare l'effetto delle classi miste rispetto a quelle composte da un unico sesso sulle prestazioni accademiche delle studentesse delle scuole superiori in Svizzera. Lo studio ha riportato che l'educazione in classi composte solo da ragazze può migliorare significativamente le loro prestazioni, soprattutto in materie come la matematica, dove spesso

esistono disparità di genere.

Tuttavia, ci sono due aspetti importanti da considerare. Il primo è che in questo scenario ci sono donne che scelgono percorsi accademici nelle discipline STEM, dimostrando quindi che l'*habitus* non determina meccanicamente le nostre scelte, ma interagisce con la nostra volontà. Pertanto, è possibile “resistere” a certi condizionamenti. Il secondo è che l'*habitus* non è immutabile, fisso. Nel corso del tempo, le donne sono riuscite ad inserirsi nei corsi di economia, che in passato erano dominati dagli uomini, analogamente a quelli di ingegneria, mentre ora sono diventati *gender neutral*, al punto che oggi non “sembra strano” vedere donne che scelgono questi percorsi. L'*habitus* rappresenta quindi il punto d'incontro fra *agency* e struttura, fra volontà dell'individuo e i condizionamenti sociali che caratterizzano l'ambiente in cui l'individuo si forma (Turnbull et al., 2019; Edgerton et al., 2014).

2.2 PENSIERO COMPUTAZIONALE E DIFFERENZE DI GENERE: EVIDENZE EMPIRICHE

Gli studi che hanno esplorato le differenze di genere nel pensiero computazionale hanno trovato risultati contrastanti. Nello studio di Kalelioğlu (2015) l'obiettivo era quello di indagare se esiste una differenza di genere nelle capacità di pensiero riflessivo delle studentesse e degli studenti della scuola primaria rispetto alla risoluzione dei problemi. I risultati indicano che l'insegnamento della programmazione tramite il sito Code.org non ha evidenziato alcuna differenza nelle capacità di pensiero riflessivo verso la risoluzione dei problemi. Tuttavia, è stato osservato un leggero incremento nelle capacità di pensiero riflessivo delle studentesse verso la soluzione dei problemi rispetto ai loro compagni maschi. Inoltre, bambine e bambini hanno sviluppato un atteggiamento positivo nei confronti della programmazione: le alunne hanno dimostrato di avere il medesimo successo degli alunni e la programmazione potrebbe far parte dei loro piani futuri. Tale osservazione è importante poiché suggerisce che, fornendo un'istruzione adeguata e accessibile a tutti, è possibile colmare il divario di genere nel campo della programmazione e del pensiero computazionale. Allo stesso modo, Papavlasopoulou et al. (2020), attraverso l'utilizzo di misure di eye-tracking hanno valutato le prestazioni di studenti e studentesse nuovi alla codifica, di età compresa tra gli 8 e 17 anni durante alcuni workshop. I risultati riferiscono di non aver trovato differenze statisticamente significative nei comportamenti di sguardo o nei guadagni di apprendimento tra ragazze e ragazzi. Tuttavia, sono emerse alcune differenze qualitative legate al genere tra ragazze e ragazzi nelle strategie utilizzate e nella percezione delle attività di codifica. Lo studio di Funke e Geldreich (2017) ha voluto esaminare le differenze di genere nei progetti di programmazione realizzati da bambine e bambini, di età compresa tra i 5 e i 12 anni, utilizzando Scratch. I risultati mostrano che non esistono differenze di genere significative nelle capacità di programmazione tra alunne e alunni della scuola primaria, supportando l'idea che, con un'adeguata istruzione e incoraggiamento entrambi i generi possono

eccellere in attività di programmazione. Lo studio di Sullivan e Bers (2016), invece, che si è concentrato su bambini di età compresa tra i 4 e i 7 anni, ha trovato prestazioni simili tra ragazze e ragazzi in compiti di CT che coinvolgono abilità di codifica di base, ma prestazioni significativamente migliori dei ragazzi nell'uso di costrutti di codifica più avanzati.

Nello studio condotto da Román-González et al. (2017) è stato messo in luce il rapporto che sussiste tra le capacità di CT e il genere in bambini spagnoli dalla quinta alla decima elementare. Dall'analisi dei dati raccolti gli autori hanno evinto come vi sia una differenza statisticamente significativa, a favore dei maschi, nei punteggi del “*Computational Thinking test*” (CTt, strumento sviluppato dagli autori per valutare le capacità di CT degli studenti). Confrontando poi i risultati ottenuti dai maschi con i risultati ottenuti dalle femmine è emerso come le medie dei punteggi dei ragazzi risultassero sempre più elevate rispetto alle medie nei punteggi del CTt ottenute dalle ragazze. Risulta inoltre rilevante l'aumento del divario tra i punteggi degli studenti e delle studentesse con l'aumentare del grado scolastico. Una revisione della letteratura di Espino e González (2016) conferma l'esistenza di differenze di genere nei processi informatici tra ragazzi e ragazze, ma sottolinea che tutti possono sviluppare abilità di pensiero computazionale correlate, così come acquisire ugualmente abilità informative.

2.3 DIFFERENZE DI GENERE NELLA PERCEZIONE DI AUTOEFFICACIA NEL PENSIERO COMPUTAZIONALE

Lo psicologo canadese Albert Bandura ha coniato il concetto di autoefficacia definendola come la convinzione di una persona di poter completare un particolare compito o svolgere un particolare ruolo all'interno di un ambito specifico (Bandura, 2006). Non è una valutazione delle effettive abilità possedute dalle persone, quanto piuttosto una convinzione su cosa si è in grado di fare con quelle abilità, dominando specifiche attività, situazioni o aspetti del funzionamento psicologico o sociale (Bandura, 1986; Caprara, 2001). Secondo l'autore il modo in cui le persone si comportano è determinato dalla loro percezione di quanto siano abili, competenti ed efficaci (Bandura, 1986). Le persone evitano le attività che credono eccedano le loro capacità di coping, ma intraprendono ed eseguono con sicurezza quelle che si ritengono capaci di gestire (Bandura, 1977). I giudizi di autoefficacia determinano anche quanti sforzi le persone spenderanno e per quanto tempo persisteranno di fronte agli ostacoli o alle esperienze avversive. Quando si trovano in difficoltà, le persone che nutrono dubbi sulle proprie capacità rallentano i loro sforzi o si arrendono del tutto, mentre coloro che hanno un forte senso di efficacia esercitano uno sforzo maggiore per superare le

sfide (Bandura & Schunk, 1981; Brown & Inouye, 1978; Schunk, 1981; Weinberg et al., 1979). Un'elevata perseveranza di solito produce risultati ad alte prestazioni.

La valutazione delle persone sulle proprie capacità influenza i loro pensieri e le loro reazioni emotive durante le interazioni con l'ambiente. Coloro che si ritengono inefficaci si concentrano sulle proprie carenze e vedono le difficoltà come più grandi di quanto siano (Beck, 1976; Lazarus & Launier, 1978; Meichenbaum, 1977; Sarason, 1975). Questo porta a stress e riduce la performance, spostando l'attenzione dalle soluzioni ai problemi. Al contrario, chi ha un forte senso di efficacia si concentra sulle richieste della situazione e viene stimolato dagli ostacoli a impegnarsi di più.

L'autoefficacia, sebbene non permetta di eseguire il compito in caso di mancanza di conoscenze e di competenze necessarie, motiva comunque la persona a migliorare le proprie capacità. Bandura (1986, 2000) sostiene che il senso di autoefficacia può essere supportato e incrementato attraverso quattro principali fonti di informazione:

1. *esperienze dirette*: la percezione di successo o di fallimento durante l'esperienza di performance in un dominio; risultano essere la fonte più potente dell'autoefficacia;
2. *esperienze vicarie*: basate sul modellamento o apprendimento osservativo; consistono nella percezione di modelli di ruolo identificabili come di successo;
3. *persuasione verbale*: l'incoraggiamento o lo scoraggiamento ricevuto dagli altri quando ci si impegna in un compito;
4. *stati fisiologici e affettivi*: l'influenza delle informazioni somatiche provenienti da stati fisiologici ed emozionali.

Inoltre, secondo Bandura (2000), l'autoefficacia ha tre dimensioni:

1. *generalità*: il grado in cui l'autoefficacia si generalizza e si estende a compiti simili
2. *forza*: grado di certezza di ottenere il risultato atteso
3. *livello*: il grado di percezione di controllo della situazione.

Gli studi che indagano la presenza di differenze di genere nell'autoefficacia nel pensiero computazionale mostrano risultati contrastanti. Uno studio di Tellhed et al. (2022) con studenti e studentesse svedesi di 12 anni ha indagato la presenza di differenze di genere nell'autoefficacia e nell'interesse verso la programmazione. I livelli di autoefficacia sono stati misurati in tre diversi tempi: all'inizio dell'anno scolastico, subito dopo lezioni di *coding* tramite programmazione a blocchi e a fine anno scolastico, con lo scopo di stabilire se i cambiamenti nell'autoefficacia fossero legati alla partecipazione alle lezioni e/o all'interesse nella programmazione. I risultati hanno mostrato un maggiore livello di autoefficacia per i maschi in tutti e tre i tempi. Nello studio di Allaire-Duquette

et al. (2022) con ragazze e ragazzi di un'età compresa tra i 10 e i 14 anni i risultati mostrano che, dopo un workshop di programmazione di due ore in un museo della scienza, le differenze di genere nell'autoefficacia per la programmazione inizialmente osservate si sono ridotte e addirittura sono scomparse. Ciò suggerisce che incoraggiare la partecipazione femminile in attività pratiche di programmazione, anche di breve durata, potrebbe essere una strategia particolarmente utile per promuovere la partecipazione delle ragazze all'informatica e offrire loro esperienze di successo. Le attività di programmazione come il *coding* hanno il potenziale per offrire alle ragazze l'opportunità di esperienze di padronanza che, attraverso il completamento con successo di compiti legati alla programmazione, possono a loro volta aumentare le convinzioni di autoefficacia delle ragazze in questo ambito (Baker, 2013). Secondo Bandura (1997), le esperienze significative di padronanza derivanti dal conseguimento enattivo di un dato ambito di funzionamento rappresentano il principale meccanismo coinvolto nella strutturazione delle convinzioni di autoefficacia sul funzionamento in quello specifico ambito. Tuttavia, ciò accade solo se la persona attribuisce la responsabilità del risultato alle proprie capacità. Ramalingam e Wiedenbeck (1998) hanno scoperto che, tra studenti universitari, non c'era una differenza sostanziale nella percezione di autoefficacia tra maschi e femmine. Sebbene i punteggi di autoefficacia delle donne fossero significativamente inferiori a quelli degli uomini dopo 12 settimane di insegnamento, questa differenza era minima, rappresentando solo il 2% della varianza.

Lo studio di Gunbatar e Karalar (2018) indaga gli effetti dell'insegnamento della programmazione utilizzando mBlock sulle percezioni di autoefficacia e sugli atteggiamenti delle studentesse e degli studenti delle scuole medie verso la programmazione. I risultati della ricerca indicano che sebbene le percezioni di autoefficacia dei ragazzi nei confronti della programmazione fossero superiori a quelle delle ragazze all'inizio della ricerca, questa differenza è stata colmata alla fine della ricerca. Quello che emerge è che l'insegnamento della programmazione con mBlock ha un impatto positivo e significativo sulle ragazze e sui ragazzi delle scuole medie, migliorando sia le loro percezioni di autoefficacia sia i loro atteggiamenti verso la programmazione, senza differenze rilevanti tra i generi.

Dall'analisi degli studi sopra citati emerge l'importanza di approfondire un aspetto importante: la valutazione delle variabili motivazionali, come la percezione di autoefficacia di bambine e di bambini, in relazione a un'attività di *coding* virtuale, che costituisce l'oggetto della presente ricerca. Nel capitolo successivo viene presentato uno studio condotto su alunne e alunni di quarta elementare, volto a indagare l'esistenza di eventuali differenze di genere nell'apprendimento e nella percezione di autoefficacia nel contesto del pensiero computazionale.

CAPITOLO III

LO STUDIO

3.1 OBIETTIVO DI RICERCA

Lo studio presentato in questa tesi si inserisce nel filone di ricerca del progetto *Computational Thinking*, promosso dal Dipartimento di Psicologia dello Sviluppo e della Socializzazione dell'Università degli Studi di Padova, in collaborazione con il Dipartimento di Matematica. Il “Progetto CoThi” ha lo scopo di introdurre bambine e bambini ai principi del pensiero computazionale tramite attività di *coding* virtuale.

Studi precedenti, appartenenti allo stesso filone di ricerca, hanno dimostrato l'efficacia di un training di *coding* in alunni di classe prima della scuola primaria (Arfè et al., 2019, 2020) e che differenze di genere nel coding a favore dei maschi possono emergere già nei primi anni della scuola primaria, verso i 5-7 anni (Montuori et al., 2022).

Sulla base di queste evidenze, il presente studio si è posto l'obiettivo di indagare la presenza di differenze di genere tra bambine e bambini di classe quarta della scuola primaria nell'apprendimento del *coding* e nell'autoefficacia legata alla programmazione, analizzando come quest'ultima vari nel tempo. La ricerca valuta un intervento effettuato su 89 bambini appartenenti a tre scuole diverse del territorio di Padova, con l'obiettivo di rispondere ai seguenti quesiti:

1. Esplorare se esistono differenze di genere nell'apprendimento del *coding* tra bambine e bambini di classe quarta della scuola primaria.
2. Indagare se esistono differenze nella percezione di autoefficacia tra bambine e bambini di classe quarta della scuola primaria prima di un intervento di *coding*
3. Esaminare se esistono differenze nella percezione di efficacia tra bambine e bambini di classe quarta della scuola primaria in seguito a un breve intervento didattico di *coding*
4. Verificare se esistono differenze di genere nella variazione dell'autoefficacia legata alla programmazione/*coding* tra bambine e bambini di classe quarta della scuola primaria.

3.2 PARTECIPANTI

La ricerca ha coinvolto un totale di 89 bambine/i, 49 femmine e 40 maschi, provenienti da cinque classi quarte di tre scuole primarie differenti del territorio di Padova, ognuna con un livello socioeconomico diverso. Le/gli studenti sono stati assegnati in modo casuale a due gruppi: 55 studenti al gruppo sperimentale (F=30, M=25) e 34 al gruppo di controllo (F=19, M=15). L'età media dei partecipanti è di 8-10 anni.

Al campione è stato consegnato il consenso informato di adesione allo studio, nel quale sono stati chiariti gli scopi della ricerca ed è stata comunicata la possibilità di ritirarsi dallo studio in qualsiasi momento. Insieme ad esso, è stato somministrato ai genitori dei partecipanti un breve questionario sociodemografico con lo scopo di stimare il loro Status Socioeconomico (SES), il loro livello di familiarità con la tecnologia (Fam Tech) e il loro background migratorio. Per il SES è stato chiesto ad entrambi i genitori di riportare il titolo di studio più alto conseguito e la professione, mentre per il Fam Tech venivano richieste informazioni riguardo alla frequenza e alla modalità di utilizzo di dispositivi digitali quali computer, tablet e smartphone da parte dei figli. Nel questionario erano anche presenti domande relative alla tipologia di computer e di mouse in possesso dal bambino. Nelle analisi effettuate in questo studio non sono state prese in considerazione queste ultime informazioni riportate dai genitori.

3.3 STRUMENTI

Il disegno sperimentale ha previsto la raccolta delle misure relative al senso di autoefficacia rispetto al *coding*/programmazione attraverso l'Elementary Student Coding Attitudes Survey (Mason & Rich, 2020), mentre per la valutazione delle abilità di *coding* è stata utilizzata l'applicazione web CoThi, una piattaforma ispirata a Code.org².

1. Elementary Student Coding Attitudes Survey (ESCAS)

La Scala ESCAS, sviluppata da Mason e Rich (2020), è stata progettata per misurare i cambiamenti negli atteggiamenti delle giovani stendesse e studenti (dalla quarta alla sesta elementare) nei confronti della programmazione. È formata da 23 item su scala Likert a 6 punti, che vanno da “per niente” (1) a “moltissimo” (6). La scala indaga cinque costrutti: *Confidence*, Interesse, Utilità Percepita, Valore Sociale e Percezione dei programmatori. La versione standardizzata è in lingua inglese e non è presente una versione validata in italiano. Infatti, per questo studio, la scala è stata interamente

² <https://code.org>

tradotta e il termine “programmazione” è stato associato al termine “*coding*”, dato che i partecipanti hanno acquisito familiarità con il termine inglese durante il corso dei training.

Lo scopo di questo questionario era capire se, dopo il training di *coding*, bambine e bambini avessero sviluppato un senso di efficacia e fiducia nelle proprie capacità di programmazione e problem-solving e un atteggiamento positivo verso l’ambito informatico. In questo studio sono stati somministrati 15 su 23 item, in particolare quelli relativi alle sotto scale: *Confidence*, Interesse e Utilità Percepita:

• *Confidence*: questo costrutto è misurato con affermazioni che valutano la fiducia degli studenti nelle loro abilità di *coding*, come la capacità di imparare a codificare e di risolvere problemi di codifica. Gli items che sono stati somministrati sono:

1. Posso imparare a programmare/fare coding;
2. Sono brava/o a programmare/fare coding;
3. Sono brava/o a risolvere problemi;
4. So scrivere istruzioni chiare per un robot o un computer;
5. Se il mio programma non funziona, so trovare l’errore e risolverlo;
6. Mi è stato detto che sono brava/o a programmare/fare coding.

• *Interesse*: è valutato come la misura in cui gli studenti trovano il coding interessante e divertente. Questo costrutto si concentra sui sentimenti positivi degli studenti verso il coding e il loro desiderio di imparare di più su di esso. Gli items somministrati sono:

7. Mi piace la programmazione/il coding, o penso che mi piacerebbe;
8. Mi piacerebbe imparare di più sulla programmazione/sul coding;
9. Risolvere problemi di programmazione/di coding sembra divertente;
10. La programmazione/il coding è interessante;
11. Mi piacerebbe studiare programmazione/coding.

• *Utilità percepita*: è misurata considerando quanto gli studenti ritengono che il coding sia utile per il loro futuro. Gli items somministrati sono:

12. Posso usare le abilità di programmazione/coding in altre materie scolastiche;
13. Sapere come programmare/fare coding mi aiuterà a creare cose utili;
14. Sapere come programmare/fare coding mi aiuterà a risolvere problemi;
15. Penso che avrò bisogno di capire la programmazione/il coding per il mio futuro lavoro.

Pagina 1 di 3



Posso imparare a programmare / fare coding.

Per niente ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6 Moltissimo

Sono bravo a programmare / fare coding.

Per niente ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6 Moltissimo

Sono bravo a risolvere problemi.

Per niente ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6 Moltissimo

So scrivere istruzioni chiare per un robot o un computer.

Per niente ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6 Moltissimo

Se il mio programma non funziona, so trovare l'errore e risolverlo.

Per niente ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6 Moltissimo

Mi è stato detto che sono bravo a programmare / fare coding.

Per niente ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6 Moltissimo

Prossima pagina

Figura 1: Item esempio scala ESCAS

2. Abilità di Coding

Le abilità di *coding* sono state valutate attraverso CoThi, un'applicazione web disegnata dal gruppo di ricerca per implementare i training e le valutazioni sul pensiero computazionale. Nella piattaforma ogni bambina e ogni bambino aveva la possibilità di accedere al proprio account selezionando il proprio nome all'interno della classe di appartenenza. Questo ci permetteva di seguire i suoi miglioramenti durante il percorso.

Le prove sono basate su un linguaggio “a blocchi”: non è richiesto di scrivere codici testuali, ma di trascinare dei blocchi che rappresentano dei comandi. Studentesse e studenti hanno il compito di creare un programma che guidi un personaggio (ad esempio l'uccellino rosso) nel conseguimento di uno o più obiettivi (ad esempio il maialino verde) attraverso il percorso più breve possibile.

Come si può vedere nella **Figura 2**, a sinistra è presente la griglia 8x8 che mostra la posizione di partenza del personaggio, la posizione dell'obiettivo e le caselle libere da ostacoli. Nella parte destra, invece, sono presenti le istruzioni, i blocchi utilizzabili e l'area di lavoro, in cui è possibile creare il codice. Nell'area di lavoro viene anche indicato il numero di blocchi ideali da utilizzare. L'esercizio è considerato risolto quando il personaggio raggiunge l'obiettivo, indipendentemente dal numero di blocchi utilizzati. Il codice viene creato trascinando e attaccando i blocchi necessari sotto al blocco “quando si clicca su Esegui”, sempre presente di default. Sotto alla griglia, infatti, è presente il tasto “Esegui” che fornisce un riscontro sul percorso che il personaggio segue a seconda dei blocchi utilizzati. Ogni volta che viene premuto il tasto “Esegui” equivale a un tentativo. Il numero massimo di tentativi è tre per ogni esercizio, incluse le prove di esempio.



Figura 2: Esempio prova I del test di coding su CoThi.it

Per ognuno di questi esercizi, CoThi registra automaticamente:

- il *tempo di pianificazione* (in secondi): dal momento in cui viene avviato l'esercizio a quando viene selezionato il primo blocco da spostare;
- il *tempo di esecuzione*: da quando viene spostato il primo blocco fino a quando viene cliccato il tasto "Esegui"
- il *numero* di tentativi utilizzati per svolgere correttamente l'esercizio;
- l'utilizzo delle *funzioni specifiche* per risolvere i vari item (ad esempio il ciclo ripeti, l'istruzione condizionale ecc., descritte nel paragrafo seguente).
- l'*accuratezza*: il punteggio assegnato ad ogni esercizio, incluse le prove di esempio, è di due punti per ogni esercizio svolto correttamente al primo tentativo, un punto per il secondo tentativo e zero punti per il terzo tentativo.

3.4 PROCEDURA

Il 1° dicembre 2023 il Comitato Etico del centro HIT ha approvato il "*Progetto CoThi*" consentendo l'inizio delle attività, che si sono svolte da gennaio a giugno 2024. Lo studio è stato condotto nelle scuole durante le ore di lezione e ha previsto diverse fasi:

- **Fase di pre-test (T1):** in questa fase, le alunne e gli alunni sono stati suddivisi in piccoli gruppi e accompagnati dal gruppo di ricerca in un'aula, messa a disposizione dalla scuola, sufficientemente spaziosa e silenziosa per garantire un ambiente adatto alla valutazione delle capacità. Qui, sia per il gruppo sperimentale che per il gruppo di controllo, sono state valutate le abilità preliminari di *coding* su pc portatile tramite la piattaforma CoThi.it. Inoltre, è stato somministrato il questionario ESCAS che indaga il senso di autoefficacia di bambine e bambini relativo al *coding*/programmazione, così come il loro interesse in quest'ultimo.

- **Fase di training per il gruppo sperimentale:** subito dopo la fase di pre-test, il gruppo sperimentale è stato sottoposto all'intervento, mentre il gruppo di controllo ha proseguito con le normali attività didattiche curricolari. Il training, della durata di un mese, si è articolato in otto incontri da un'ora l'uno, svolti collettivamente nell'aula informatica del plesso scolastico, dotata di computer fissi (a disposizione per ogni alunno) e proiettore. L'obiettivo di questa fase è stato quello di potenziare le abilità di pensiero computazionale. All'inizio di ogni incontro venivano descritti brevemente gli obiettivi e le modalità di utilizzo di eventuali nuovi blocchi. Gli esercizi si differenziavano ad ogni lezione, a seconda della diversa funzione che gli alunni dovevano apprendere. Il protocollo prevedeva l'insegnamento diversi concetti chiave come: 1) l'esecuzione sequenziale, 2) il ciclo "ripeti", 3) il debugging, 4) l'esecuzione condizionale e 5) il concetto di funzione.

1) L'*esecuzione sequenziale* di una serie di istruzioni prevede che i programmi vengano eseguiti un'istruzione alla volta, in un ordine ben preciso (**Figura 3**).

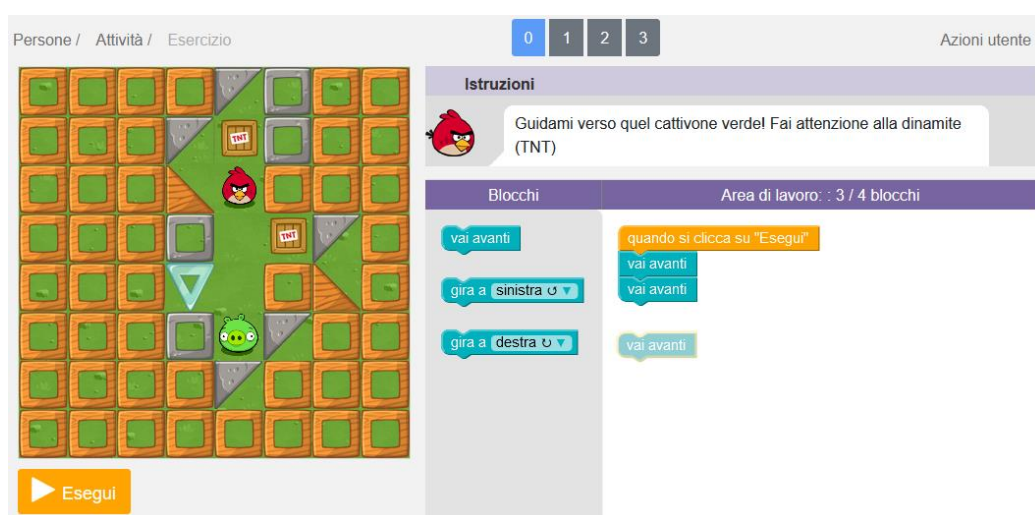


Figura 3: Esercizio 0 dell'incontro I

2) Il ciclo “*ripeti*” permette di sintetizzare un algoritmo (soluzione) in una sequenza di azioni più breve (**Figura 4**).



Figura 4: Esercizio 0 dell'incontro II

3) Il *debugging*, invece, è l'abilità che permette alla/o studente di identificare la presenza di eventuali errori all'interno di un algoritmo, con l'obiettivo di correggerlo (**Figura 5**).



Figura 5: Esercizio 0 incontro IV

4) Le *istruzioni condizionali* permettono ad un programma di eseguire diverse sequenze di istruzioni in base al verificarsi di determinate situazioni. (Figura 6).

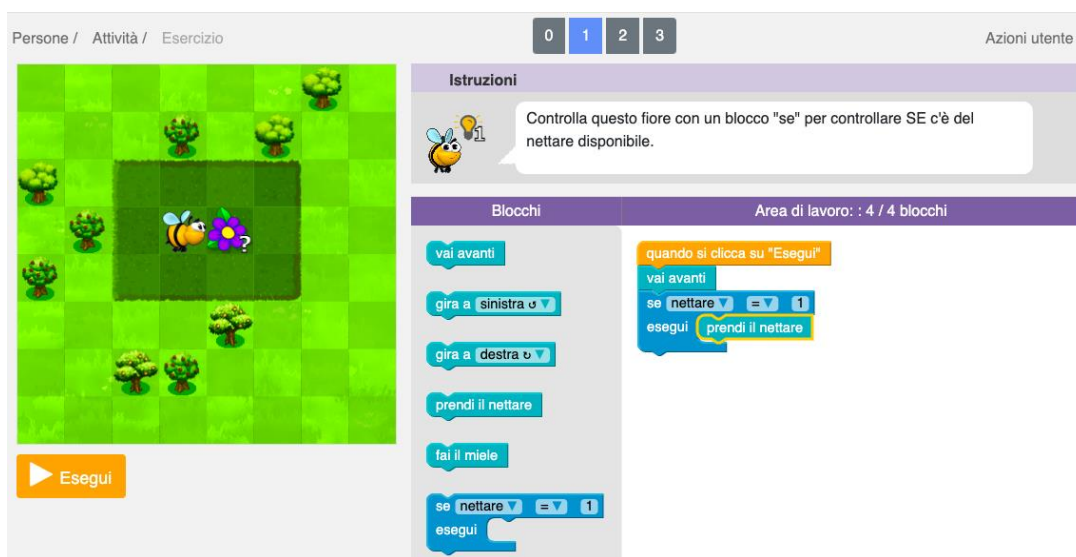


Figura 6: *Esercizio 1 incontro V*

5) Infine, il concetto di *funzione* si riferisce ad una sequenza di istruzioni che viene definita ed etichettata con un nome (Figura 7).



Figura 7: *Esercizio 1 incontro VII*

Bambine e bambini sono stati incoraggiati a risolvere gli esercizi individualmente, leggendo sempre le istruzioni e riflettendo su come utilizzare il minor numero possibile di blocchi. In questa fase non era previsto un numero massimo di tentativi. In caso di necessità, è stato fornito un supporto ai bambini, dando suggerimenti e strategie per risolvere gli esercizi, senza

mai offrire una soluzione diretta ma utilizzando una strategia di scaffolding, aiutando i bambini a raggiungere la soluzione in autonomia. Dopo il superamento delle prove, soprattutto quelle più complesse o che richiedevano l'utilizzo di nuovi blocchi, veniva lasciato spazio a una discussione collettiva, in cui alcuni bambini descrivevano alla classe il modo in cui avevano risolto l'esercizio. L'obiettivo era quello di mettere in evidenza che per uno stesso problema si potevano trovare strategie di risoluzioni differenti, ma anche portare i bambini a trovare insieme la soluzione più efficiente, nel caso in cui nessuno l'avesse trovata autonomamente.

- **Fase di post-test (T2):** dopo la fase di training per il gruppo sperimentale, si è svolta la fase T2 in cui, con le stesse modalità del T1, il gruppo sperimentale e di controllo hanno svolto i test di valutazione delle abilità di *coding* seguiti dal questionario ESCAS. Questa seconda fase di test risulta importante per la dimostrazione dell'efficacia dell'intervento.
- **Fase di training per il gruppo di controllo:** a seguito della fase di post-test, anche il gruppo di controllo è stato sottoposto al training nelle stesse modalità del gruppo sperimentale, così da poter beneficiare degli effetti dovuti al training. Nel frattempo, il gruppo sperimentale ha proseguito con le normali attività didattiche.

Tabella 1: Disegno sperimentale relativo al gruppo sperimentale

GRUPPO SPERIMENTALE		
Prima dell'intervento (T1)	Training (1 mese)	Dopo l'intervento (T2)
ESCAS + Coding	8 ore di lezione, 2 lezioni un'ora alla settimana	ESCAS + Coding

Tabella 2: Disegno sperimentale relativo al gruppo di controllo

GRUPPO DI CONTROLLO			
Prima dell'intervento (T1)	1 mese	Prima dell'intervento (T2)	Training (1 mese)
ESCAS + Coding	Consuete attività didattiche	ESCAS + Coding	8 ore di lezione, 2 lezioni un'ora alla settimana

Nelle fasi di valutazione T1 e T2 sono state proposte a bambine e bambini le stesse attività con lo scopo di osservare se fosse presente un miglioramento.

CAPITOLO IV

RISULTATI

4.1 ANALISI PRELIMINARI

Prima di condurre le analisi che consentiranno di rispondere alle domande di ricerca poste nei capitoli precedenti, sono state effettuate delle esplorazioni preliminari sul gruppo sperimentale e sul gruppo di controllo con l'obiettivo di verificare la presenza di eventuali differenze statisticamente significative in termini di età e familiarità con la tecnologia (FamTech). Poiché il test di Shapiro-Wilk ha rilevato che i dati relativi alle variabili considerate non seguono una distribuzione normale, i confronti tra gruppi sono stati eseguiti utilizzando il test non parametrico di Wilcoxon-Mann-Whitney. Come illustrato in **Tabella 3**, i risultati per il campione totale, che confrontano la condizione sperimentale con quella di controllo, non evidenziano differenze statisticamente significative per età e FamTech. I p-value ottenuti, infatti, sono superiori al livello di significatività convenzionale di 0.05.

Tabella 3: Test di Wilcoxon-Mann-Whitney per il gruppo sperimentale e il gruppo di controllo per le variabili età e familiarità con la tecnologia

	CAMPIONE TOTALE		GRUPPO SPERIMENTALE		GRUPPO DI CONTROLLO	
VARIABILI	Z	p-value	Z	p-value	Z	p-value
Età (mesi)	0.64	0.52	1.38	0.17	- 0.22	0.83
FamTech	- 0.90	0.37	- 0.31	0.76	1.21	0.23

Successivamente, è stata valutata l'omogeneità all'interno di ciascuna condizione (sperimentale e di controllo), confrontando maschi e femmine in relazione alle variabili età e familiarità con la tecnologia. L'analisi ha rivelato, anche in questo caso, che non sono presenti differenze statisticamente significative per le due variabili, suggerendo che entrambe le condizioni sono comparabili rispetto a queste caratteristiche. Pertanto, possiamo affermare che i due gruppi partono da uno stato iniziale simile per quanto riguarda l'età e il FamTech. Questa omogeneità è cruciale per assicurare la validità delle analisi successive, riducendo il rischio che differenze nei risultati osservati possano essere attribuite a queste variabili piuttosto che agli interventi o alle condizioni sperimentali.

Infine, è stata condotta un'ulteriore analisi preliminare per verificare se l'intervento è stato efficace e se bambine e bambini apprendono in egual modo in seguito ad un training di *coding*. Dai valori riportati in **Tabella 4** è possibile notare che il delta (differenza tra post-test e pre-test) dei maschi del gruppo sperimentale tende ad essere maggiore rispetto a quello delle femmine suggerendo che la fase di training ha contribuito in modo rilevante nel miglioramento di queste abilità negli alunni. Anche

le femmine appartenenti alla medesima condizione hanno registrato una differenza tra le medie delle prestazioni superiore rispetto alle bambine che sono state sottoposte solo successivamente alla fase di training (gruppo di controllo). Tuttavia, è possibile notare che nella differenza tra le media del post-test e del pre-test i maschi tendono ad avere un miglioramento maggiore rispetto alle femmine con una differenza di 2.19 nel gruppo sperimentale e di 2.74 nel gruppo di controllo, mostrando che gli alunni presentano un maggiore apprendimento rispetto alle alunne. Il test non parametrico di Wilcoxon-Mann-Whitney conferma la differenza di genere riscontrata precedentemente nel miglioramento da T1 a T2: per il gruppo sperimentale, che ha partecipato al training di *coding*, è stato prodotto un valore Z di 2.51 e un p-value significativo di 0.0121 (considerato significativo se < 0.05), indicando una differenza statisticamente significativa nell'apprendimento tra maschi e femmine. Questo risultato è confermato anche dal test di Kruskal-Wallis, il quale riporta un valore di chi-quadrato di 0.0118, anch'esso significativo a livello statistico. Allo stesso modo, nel gruppo di controllo, che ha svolto esercizi di *coding* unicamente durante le valutazioni, il test di Wilcoxon-Mann-Whitney ha rilevato un valore Z di 2.09 e un p-value significativo di 0.0363, suggerendo differenze nell'apprendimento tra maschi e femmine nel miglioramento da T1 a T2. Questa differenza statisticamente significativa è a favore degli alunni, che hanno ottenuto una media di 4.27 rispetto alla media di 1.53 delle alunne. Anche in questo caso, il test di Kruskal-Wallis ha prodotto un valore di chi-quadrato di 0.0348, risultando significativo a livello statistico. In conclusione, i risultati suggeriscono che l'intervento di *coding* è stato efficace e che sono presenti delle differenze di genere significative tra alunne e alunni nell'apprendimento.

Tabella 4: Media, deviazione standard e test di Wilcoxon-Mann-Whitney del gruppo sperimentale e di controllo diviso per genere per la variabile apprendimento coding a T1, T2 e del loro delta

	GRUPPO SPERIMENTALE						GRUPPO DI CONTROLLO					
	Maschi (n = 25)		Femmine (n = 30)		Test di Wilcoxon- Mann-Whitney		Maschi (n = 15)		Femmine (n = 19)		Test di Wilcoxon- Mann-Whitney	
VARIABILI	M	SD	M	SD	Z	p-value	M	SD	M	SD	Z	p-value
Accuratezza coding T1	3.92	3.71	3.57	3.47	0.26	0.80	4.33	3.46	4.16	3.85	0.05	0.96
Accuratezza coding T2	9.08	2.66	6.53	3.68	2.58	0.0100*	8.60	2.23	5.68	3.65	2.27	0.0231*
Delta accuratezza T2-T1	5.16	3.13	2.97	3.00	2.51	0.0121*	4.27	3.63	1.53	3.69	2.09	0.0363*

Il **Grafico 1** fornisce una rappresentazione visiva dell'apprendimento del *coding* di bambine e bambini, calcolato come la differenza (delta) tra i risultati del post-test e del pre-test, suddiviso per gruppo. Da esso emerge che, sia per il gruppo sperimentale che per il gruppo di controllo, i maschi tendono ad avere una mediana (linea orizzontale all'interno della scatola che indica il valore centrale della distribuzione) superiore rispetto alle femmine al post-test, ma non al pre-test. Questo indica che gli alunni ottengono, in media, un miglioramento nei punteggi che risultano più elevati rispetto ai

punteggi ottenuti dalle alunne nelle prove di *coding*. In particolare, nel gruppo di controllo, ciò suggerisce che, anche senza la partecipazione al training, i maschi tendono ad avere una migliore prestazione nel *coding* al T2 rispetto alle femmine.

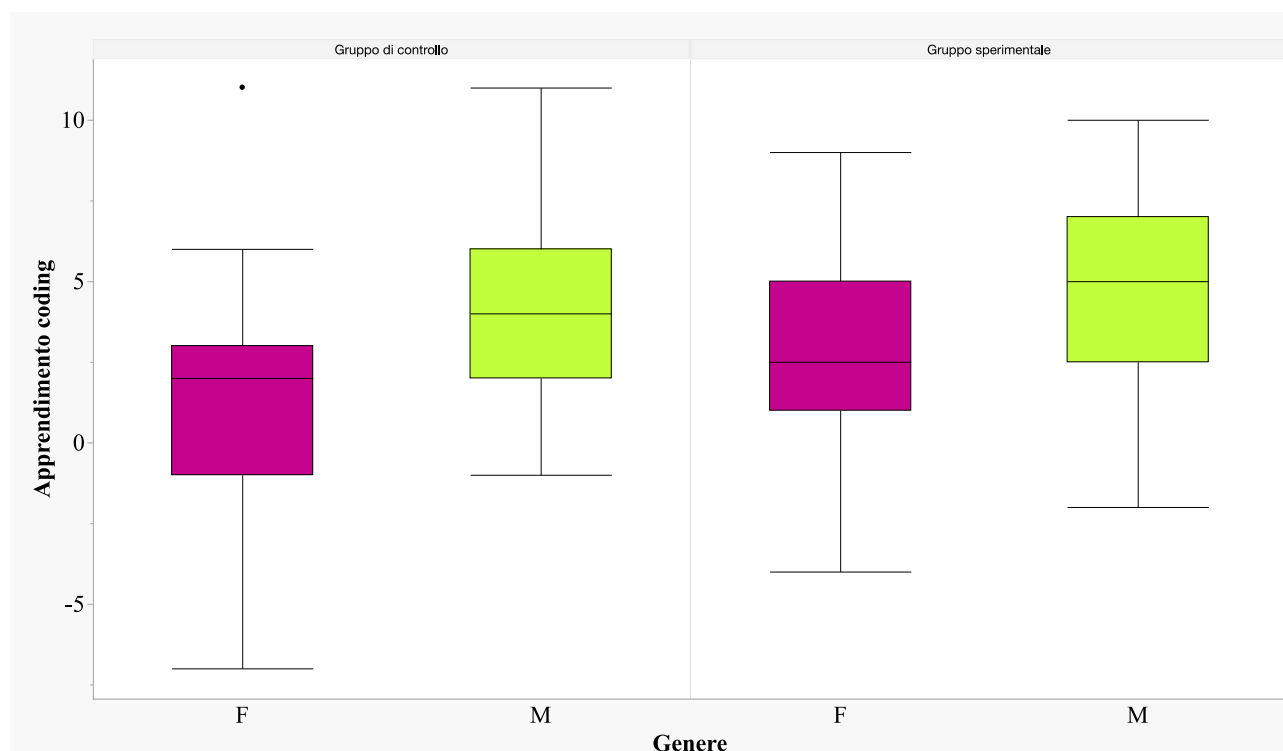


Grafico 1: *Apprendimento del coding, confronto per genere e per gruppo*

In conclusione, i risultati confermano l'efficacia del training, il quale ha prodotto un miglioramento maggiore nell'apprendimento di bambine e bambini del gruppo sperimentale rispetto al gruppo di controllo. Inoltre, il test di Wilcoxon-Mann-Whitney rivela differenze statisticamente significative tra maschi e femmine in entrambi i gruppi a T2, con gli alunni che hanno ottenuto un miglioramento e punteggi medi più alti rispetto alle alunne. La differenza di genere appare più pronunciata nel gruppo sperimentale, evidenziato da Z più alto (2.51) e p-value più basso (0.0121). L'elevata deviazione standard, inoltre, suggerisce che esistono notevoli differenze individuali all'interno di ciascun gruppo di genere a causa dell'ampia distribuzione dei punteggi intorno alla media del gruppo.

4.2 DIFFERENZE DI GENERE NELL' AUTOEFFICACIA AL PRE-TEST

In questi ultimi sottoparagrafi, vengono riportate le analisi relative all'autoefficacia. In primo luogo, sono stati valutati i punteggi ottenuti da bambine e bambini di entrambe le condizioni (sperimentale e controllo) nella scala dell'autoefficacia nel *coding*/programmazione (ESCAS) al T1, inclusi i punteggi delle sottoscale di *Confidence*, *Interest* e *Utility*. Queste analisi hanno l'obiettivo di rispondere alla domanda di ricerca che indaga l'eventuale presenza di differenze di genere

nell'autoefficacia legata alla programmazione tra alunne e alunni di quarta elementare prima di un intervento di *coding*.

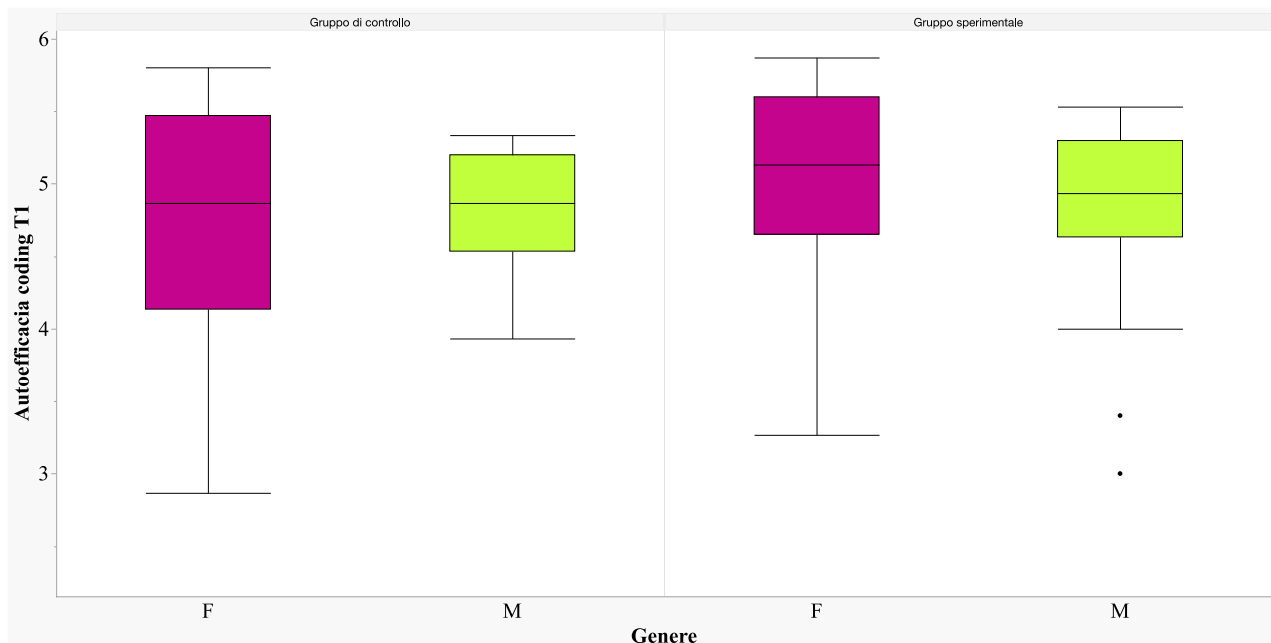


Grafico 2: Autoefficacia nel Coding al T1, confronto per genere e per gruppo

Il **Grafico 2** fornisce una rappresentazione visiva dei livelli di autoefficacia delle alunne e degli alunni, suddivisi per gruppo sperimentale e di controllo, prima dell'intervento di *coding* (T1). Da esso si può osservare che le femmine tendono ad avere una mediana leggermente superiore rispetto ai maschi in entrambe le condizioni, suggerendo una maggiore percezione di autoefficacia prima della partecipazione al training. La maggiore variabilità nei punteggi di autoefficacia nel *coding* delle femmine rispetto ai maschi, visibile attraverso una distribuzione più ampia nel grafico, potrebbe indicare una gamma più diversificata di esperienze o percezioni individuali in questo ambito tra le alunne. Inoltre, dal grafico, si osserva la presenza di alcuni outliers (tra le femmine e tra i maschi del gruppo sperimentale) che indicano alcuni casi particolari che deviano significativamente dalla tendenza generale.

Per approfondire questi risultati, è stato eseguito il test di Wilcoxon-Mann-Whitney dal momento che le variabili considerate non seguono una distribuzione normale. Per il gruppo sperimentale, il test, che ha confrontato l'autoefficacia nel coding di maschi e femmine a T1, ha restituito un valore Z di -1.27 e un p-value di 0.20, mentre per il gruppo di controllo il valore Z è di -0.12 con un p-value di 0.90. Entrambi i p-value sono superiori al livello di significatività di 0.05 indicano l'assenza di differenze statisticamente significative tra maschi e femmine al pre-test (T1). Valori elevati come questi suggeriscono che le differenze osservate potrebbero essere dovute al caso piuttosto che a differenze reali tra i gruppi. Pertanto, eventuali differenze riscontrate nei risultati del post-test possono

essere attribuite principalmente all'intervento stesso, piuttosto che a differenze preesistenti tra i gruppi.

Rispetto alla prestazione al pre-test, il gruppo sperimentale presenta per i maschi un valore medio di 4.83 (SD = 0.62), mentre per le femmine è di 5.02 (SD = 0.66). Nel gruppo di controllo, i maschi hanno una media di 4.79 (SD = 0.45), e le femmine una media di 4.76 (SD = 0.80). Da questi dati si osserva che il gruppo sperimentale ha medie leggermente superiori rispetto al gruppo di controllo per entrambi i sessi, con le alunne del gruppo sperimentale che presentano una media più elevata rispetto agli alunni per quanto riguarda l'autoefficacia nella programmazione. Inoltre, il gruppo sperimentale mostra deviazioni standard più elevate, indicando una maggiore variabilità nei punteggi rispetto al gruppo di controllo, confermando quanto osservato nel grafico.

Infine, dalla **Tabella 5** possiamo notare che, in entrambi i gruppi tra le sottoscale non sono presenti differenze statisticamente significative tra maschi e femmine a T1 in nessuna delle tre variabili, ad eccezione della sottoscala *Confidence* del gruppo sperimentale, che presenta un valore Z di - 1.78 e un p-value associato di 0.0749. Sebbene la differenza non risulti statisticamente significativa, il p-value si avvicina al livello convenzionale di significatività di 0.05. Pertanto, è importante considerare questo risultato come indicativo di una tendenza potenzialmente significativa.

Tabella 5: Media, deviazioni standard e test di Wilcoxon-Mann-Whitney del gruppo sperimentale e di controllo per la variabile autoefficacia nel coding, diviso per gruppo e per genere, al pre-test.

	GRUPPO SPERIMENTALE						GRUPPO DI CONTROLLO					
	Maschi (n = 25)		Femmine (n = 30)		Test Wilcoxon- Mann-Whitney		Maschi (n = 15)		Femmine (n = 19)		Test Wilcoxon- Mann-Whitney	
VARIABILI	M	SD	M	SD	Z	p-value	M	SD	M	SD	Z	p-value
<i>Autoefficacia Coding T1</i>	4.83	0.62	5.02	0.66	- 1.27	0.20	4.79	0.45	4.76	0.80	- 0.12	0.90
<i>Confidence T1</i>	4.35	0.69	4.66	0.80	- 1.78	0.07	4.20	0.68	4.05	0.97	0.47	0.64
<i>Interest T1</i>	5.31	0.67	5.51	0.67	- 1.29	0.20	5.52	0.45	5.49	0.56	- 0.16	0.87
<i>Utility T1</i>	4.93	0.85	4.94	0.90	- 0.15	0.88	4.77	0.56	4.89	1.03	- 1.12	0.26

4.3 DIFFERENZE DI GENERE NELL'AUTOEFFICACIA AL POST-TEST

Le analisi successive si sono concentrate sulle differenze di genere nell'autoefficacia legata al coding/programmazione al post-test (T2) per indagare se le alunne e gli alunni si percepiscono più competenti dopo l'intervento di coding. La valutazione è stata condotta principalmente sul gruppo sperimentale per verificare un eventuale effetto del training sull'autoefficacia di alunne e di alunni.

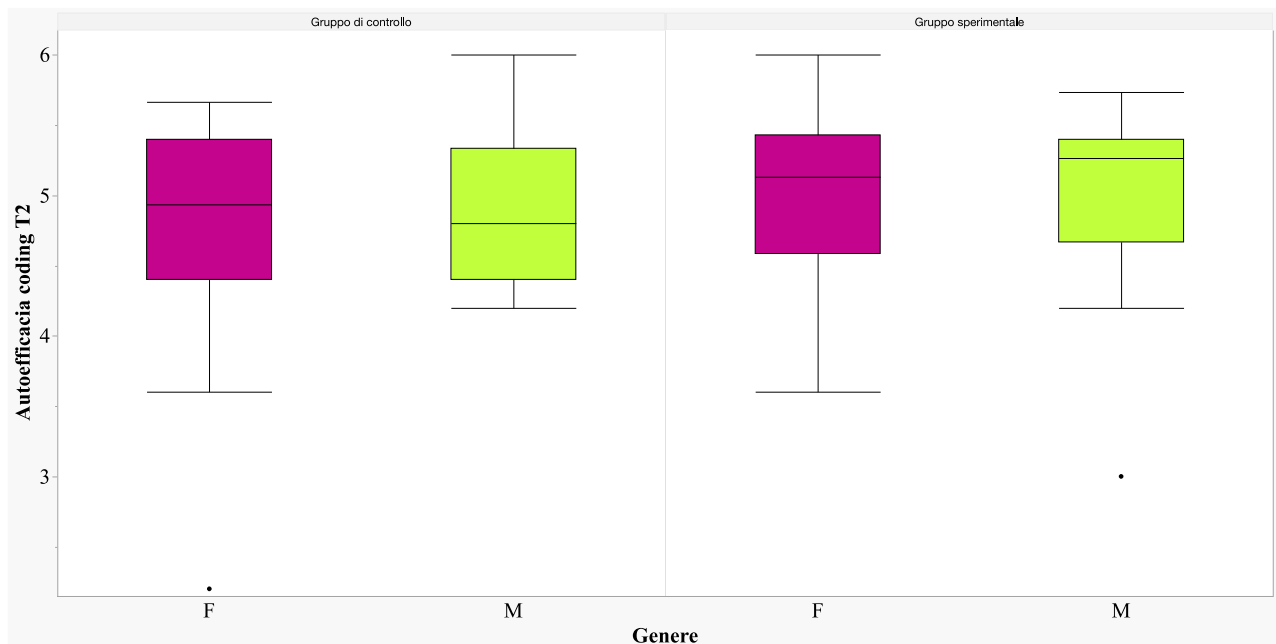


Grafico 3: Autoefficacia nel Coding al T2, confronto per genere e per gruppo

Il **Grafico 3** mostra i livelli di autoefficacia di bambine e bambini, suddivisi per gruppo di controllo e sperimentale, al post-test (T2). Da esso si può osservare che nella condizione di controllo la mediana dell'autoefficacia delle femmine è superiore a quella dei maschi. Questo indica che, senza il training, le alunne tendono ad avere una maggiore autoefficacia nella programmazione rispetto agli alunni. Nella condizione sperimentale, invece, si osserva una lieve differenza a favore dei maschi. Anche in questo caso sono presenti degli outliers (nelle femmine del gruppo di controllo e nei maschi del gruppo sperimentale) che indicano alcuni casi particolari che deviano significativamente dalla tendenza generale. Per approfondire questi risultati, è stato eseguito il test di Wilcoxon dal momento che le variabili considerate non seguono una distribuzione normale. Per il gruppo sperimentale, il test, che ha confrontato l'autoefficacia nel *coding* di maschi e femmine a T2, ha prodotto un valore Z di 0.22 e un p-value di 0.83, mentre per il gruppo di controllo il valore Z è di - 0.12 con un p-value di 0.90. Anche in questo caso, entrambi i risultati indicano l'assenza di differenze statisticamente significative tra maschi e femmine, poiché i valori ottenuti sono superiori al livello di significatività di 0.05. Rispetto alle medie relative alle percezioni di efficacia, il gruppo sperimentale presenta per i maschi un valore medio di 5.02 (SD = 0.62), mentre per le femmine è di 5.04 (SD = 0.55). Nel gruppo di controllo, i maschi hanno una media di 4.89 (SD = 0.52), e le femmine una media di 4.76 (SD = 0.83). Da questi dati emerge che il gruppo sperimentale continua a mostrare medie leggermente superiori rispetto al gruppo di controllo in ambedue i generi.

Dalla **Tabella 6** possiamo notare che, in entrambi i gruppi tra le sottoscale, non sono presenti differenze statisticamente significative in nessuna delle tre variabili. Nella sottoscala *Utility*, che indaga se il *coding* sia utile per il loro futuro, le alunne sembrano avere una media più alta rispetto ai

loro compagni maschi in entrambe le condizioni. Tuttavia, il test non parametrico non rileva disparità rilevanti dal punto di vista statistico tra i generi all'interno di ciascun gruppo. Si può quindi concludere che il training o la sola partecipazione alle valutazioni ha aumentato la percezione di autoefficacia dei bambini e delle bambine come evidenziato dall'incremento delle medie in entrambe le condizioni.

Tabella 6: Media, deviazioni standard e test di Wilcoxon-Mann-Whitney del gruppo sperimentale e di controllo per la variabile autoefficacia nel coding, diviso per gruppo e per genere al post-test.

	GRUPPO SPERIMENTALE						GRUPPO DI CONTROLLO					
	Maschi (n = 25)		Femmine (n = 30)		Test Wilcoxon- Mann-Whitney		Maschi (n = 15)		Femmine (n = 19)		Test Wilcoxon- Mann-Whitney	
VARIABILI	M	SD	M	SD	Z	p-value	M	SD	M	SD	Z	p-value
<i>Autoefficacia Coding T2</i>	5.02	0.62	5.04	0.55	0.22	0.83	4.89	0.52	4.76	0.83	- 0.12	0.90
<i>Confidence T2</i>	4.73	0.63	4.71	0.80	- 0.00	1.00	4.52	0.79	4.18	0.87	0.98	0.33
<i>Interest T2</i>	5.34	0.87	5.37	0.64	0.23	0.82	5.32	0.57	5.29	0.96	- 0.67	0.50
<i>Utility T2</i>	5.06	0.75	5.13	0.56	0.05	0.96	4.92	0.78	4.97	0.92	- 0.44	0.66

4.4 DIFFERENZE DI GENERE NELLA VARIAZIONE DELL'AUTOEFFICACIA

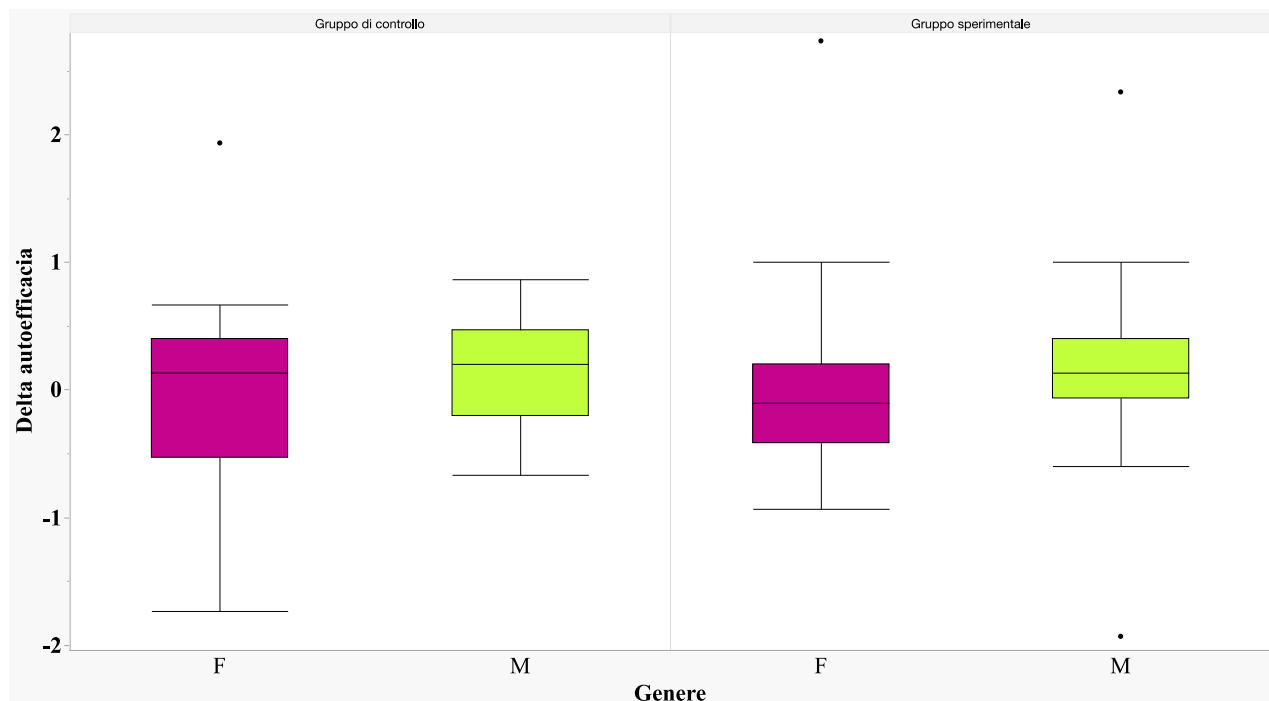


Grafico 4: Delta autoefficacia, confronto per genere e per gruppo

Il **Grafico 4** fornisce una rappresentazione visiva della differenza (delta) nella percezione di autoefficacia di alunne e alunni a seguito dell'intervento di *coding* per il gruppo sperimentale e delle due valutazioni per il gruppo di controllo. Ciò che si può osservare è che, in entrambe le condizioni, le mediane dei maschi tendono ad essere leggermente superiori a quelle delle femmine, anche se nel gruppo di controllo sembrano essere molto simili. Oltre a questo, nel gruppo che non ha partecipato al training, le alunne presentano una maggiore variabilità della distribuzione dei punteggi rispetto ai maschi. Allo stesso modo nel gruppo sperimentale sembra essere presente una maggior variabilità dei punteggi delle alunne anche se si presenta molto simile a quella degli alunni. Dal grafico è anche possibile osservare la presenza di outliers che suggeriscono casi isolati che deviano dalla tendenza generale. Il test di Wilcoxon-Mann-Whitney ha evidenziato, nel gruppo sperimentale, una differenza statisticamente significativa tra maschi e femmine per quanto riguarda il delta (T2-T1) della scala dell'autoefficacia (ESCAS) e della sottoscala *Confidence*. Dalla **Tabella 7** si evince che la differenza riscontrata nell'ESCAS è a favore degli alunni, con una media di 0.20 (SD = 0.70), un valore Z di 2.01 e un p-value significativo di 0.0442. Questa differenza viene confermata anche dal test di Kruskal-Wallis il quale riporta un valore di chi-quadrato di 0.0433, significativo a livello statistico. Per quanto riguarda la sottoscala *Confidence*, la differenza è a favore delle femmine, con una media di 0.50 (SD = 0.96), un valore Z di 2.19 e un p-value significativo di 0.0293. Anche in questo caso il test di Kruskal-Wallis ha confermato questo risultato con un valore di chi-quadrato significativo di

0.0287. Questo indica, che sebbene i maschi tendano ad avere un miglioramento nei punteggi di autoefficacia più elevato rispetto alle femmine a seguito del training di *coding*, quest'ultimo ha un impatto maggiore sulle alunne in termini di *Confidence*. Nel gruppo di controllo, non sono state osservate differenze significative ($p\text{-value} > 0.05$) tra maschi e femmine per nessuna delle variabili considerate, indicando che l'assenza di partecipazione al training non comporta variazioni significative nei punteggi di autoefficacia, utilità, interesse e fiducia.

In generale, si può concludere che l'autoefficacia, indipendentemente dal genere, è moderatamente aumentata. Questo suggerisce che il solo coinvolgimento in attività di *coding* ha un effetto positivo generale sull'autoefficacia delle alunne e degli alunni, sebbene vi siano differenze specifiche tra maschi e femmine.

Tabella 7: Media, deviazioni standard e test di Wilcoxon-Mann-Whitney del gruppo sperimentale e di controllo per la variabile delta autoefficacia nel coding, diviso per gruppo e per genere

	GRUPPO SPERIMENTALE						GRUPPO DI CONTROLLO					
	Maschi (n= 25)		Femmine (n= 30)		Test Wilcoxon		Maschi (n= 15)		Femmine (n= 19)		Test Wilcoxon	
VARIABILI	M	SD	M	SD	Z	p-value	M	SD	M	SD	Z	p-value
<i>Delta Escas</i>	0.20	0.70	0.02	0.71	2.01	0.0442*	0.10	0.44	0.01	0.74	0.67	0.50
<i>Delta Utility</i>	0.13	0.82	0.18	0.93	- 0.39	0.70	0.15	0.60	0.08	1.00	0.61	0.54
<i>Delta Interest</i>	0.03	1.01	- 1.33	0.67	1.42	0.16	- 2.00	0.55	- 0.20	0.89	- 0.16	0.87
<i>Delta Confidence</i>	0.38	0.76	0.50	0.96	2.18	0.0293*	0.32	0.89	0.13	0.83	0.54	0.59

DISCUSSIONE

Il *coding* è stato introdotto recentemente nella scuola primaria come strumento per insegnare le abilità del pensiero computazionale (Arfé et al., 2019). Numerose ricerche sostengono che la programmazione visiva basata su blocchi, come Scratch (Resnick et al., 2009), Code.org (Kalelioğlu, 2015) o CHERP (Creative Hybrid Environment for Robotics Programming, Bers et al., 2014), sia l'opzione più efficace per insegnare la programmazione ai bambini dalla scuola dell'infanzia alla scuola primaria (Sáez-López et al., 2016). Alla luce di questi sviluppi, questa ricerca si propone di analizzare se l'utilizzo dell'applicazione web CoThi, una piattaforma ispirata a Code.org, possa evidenziare differenze di genere nell'apprendimento e nella percezione di autoefficacia di alunne e di alunni di classe quarta della scuola primaria. Sebbene vi siano molte iniziative volte ad insegnare il *coding* ai giovani, esistono pochi strumenti per valutare l'effetto di tale istruzione sulle studentesse e sugli studenti. In particolare, Mason e Rich (2020) sottolineano la necessità di comprendere l'impatto affettivo dell'apprendimento della programmazione.

Analisi preliminari

Le analisi preliminari hanno, in primo luogo, evidenziato che nel campione e all'interno dei singoli gruppi (sperimentale e di controllo), non ci sono differenze statisticamente significative in termini di età e familiarità con la tecnologia (FamTech). I p-value superiori al livello di significatività convenzionale di 0.05 confermano l'omogeneità del campione rispetto a queste variabili. La mancanza di differenze significative tra i gruppi per età e FamTech è fondamentale per garantire che le eventuali variazioni nei risultati successivi possano essere attribuite all'intervento piuttosto che a differenze preesistenti nel campione.

In secondo luogo, le analisi preliminari hanno valutato l'efficacia del training e la presenza di eventuali differenze di genere nell'apprendimento. I risultati evidenziano che entrambi i gruppi mostrano un miglioramento nell'accuratezza, sebbene siano emerse delle differenze di genere statisticamente significative a T2 che non erano evidenti a T1. Per il gruppo di controllo, questo suggerisce che anche solo l'aver partecipato ad un progetto di *coding* ed essere stato sottoposto ad una prima valutazione possa portare ad avere una differenza tra i generi. I dati riportano che i maschi tendono ad ottenere punteggi più elevati nelle prove di *coding* rispetto alle femmine, sia nel gruppo sperimentale che in quello di controllo, con un miglioramento più marcato nelle performance post-test degli alunni. Le analisi, effettuate utilizzando il test di Wilcoxon-Mann-Whitney e il test di Kruskal-Wallis, confermano queste osservazioni con p-value che indicano significatività statistica.

Negli anni sono state condotte numerose ricerche sulle differenze di genere nell'apprendimento del *coding* con risultati spesso contrastanti. Ad esempio, Román-González et al. (2017) hanno

evidenziato un vantaggio a favore dei maschi tra bambini spagnoli dalla quinta alla decima elementare, mentre Funke e Geldreich (2017) non hanno riscontrato differenze di genere significative nelle capacità di programmazione tra alunne e alunni dai 5 ai 12 anni. I risultati riscontrati in questo studio sembrano confermare le conclusioni di Román-González et al. (2017). Tuttavia, come riportano Montuori et al. (2022), le differenze riscontrate possono derivare sia da differenze innate nelle abilità cognitive, che influenzano di conseguenza le performance in questo ambito, sia fattori socioculturali, come le disuguaglianze nei sistemi sociali e educativi e gli stereotipi di ruolo di genere. Pertanto, mentre i risultati del presente studio suggeriscono un vantaggio a favore dei maschi, è cruciale considerare la possibilità che tali differenze possano essere il risultato di una combinazione di fattori cognitivi e socioculturali. Un'analisi più approfondita delle influenze di questi elementi potrebbe fornire una comprensione più completa delle dinamiche in gioco nell'apprendimento del *coding* e contribuire a sviluppare interventi più mirati per ridurre il divario di genere.

Differenze di genere nella percezione di autoefficacia

Lo studio si è poi concentrato sulla percezione di autoefficacia di alunne e alunni prima e dopo un intervento di *coding*. La letteratura su questo argomento presenta risultati dissimili: ad esempio, Tellhed et al. (2022) hanno riscontrato che i maschi mostrano livelli di autoefficacia più elevati rispetto alle femmine nelle loro capacità di programmazione e nelle discipline STEM mentre Gunbatar e Karalar (2018), nella loro indagine sugli effetti dell'insegnamento della programmazione, utilizzando mBlock, sulle percezioni di autoefficacia e sugli atteggiamenti delle studentesse e degli studenti delle scuole medie, hanno stabilito che sebbene, all'inizio della ricerca, le percezioni di autoefficacia dei ragazzi nei confronti della programmazione fossero superiori a quelle delle ragazze, questa differenza veniva colmata alla fine della ricerca. Contrariamente a questi risultati, la presente ricerca non ha evidenziato differenze statisticamente significative tra i generi, né prima (T1) né dopo (T2) l'intervento di *coding*. Questo dato risulta particolarmente interessante in quanto suggerisce che l'intervento potrebbe avere un impatto omogeneo su alunni e alunne, indipendentemente dal genere, diversamente da quanto ipotizzato da parte della letteratura esistente. L'assenza di differenze di genere nelle percezioni di autoefficacia potrebbe indicare che interventi mirati, come quello di *coding* proposto, possono contribuire a ridurre o eliminare i divari di autoefficacia tra ragazzi e ragazze, fornendo uno spunto di riflessione importante per futuri studi in ambito educativo.

Tuttavia, esplorando le possibili differenze di genere nella variazione dell'autoefficacia a seguito dell'intervento, è emersa una differenza a livello statistico tra alunne e alunni nel gruppo sperimentale. In particolare, per quanto riguarda l'autoefficacia generale, è stata riscontrata una differenza significativa a favore dei maschi, suggerendo che il training di *coding* potrebbero avere un

impatto più rilevante sugli alunni. Tuttavia, nella sottoscala *Confidence*, la differenza è a favore delle femmine, indicando che l'intervento ha contribuito ad accrescere la loro fiducia nelle proprie abilità di *coding*. Nel gruppo di controllo, invece, non è emersa nessuna differenza rilevante a livello statistico.

LIMITI, PROSPETTIVE FUTURE E CONCLUSIONE

Uno dei possibili limiti di questo studio è la dimensione del campione, che potrebbe non essere rappresentativa della popolazione. Ricerche future potrebbero ampliare il gruppo di riferimento, coinvolgendo più classi di pari grado di istruzione, per verificare la generalizzabilità dei risultati. Le difficoltà linguistiche (bambine e bambini non italofofoni) riscontrate, soprattutto in due dei tre plessi, hanno rappresentato un altro possibile ostacolo alla piena comprensione, da parte di bambine e bambini, dei materiali e degli esercizi di *coding*. Questo fattore è particolarmente rilevante in contesti-classe con un'alta percentuale di studenti non italofofoni. Sicuramente un ulteriore studio sul campo necessiterebbe di un'indagine più strutturata del contesto linguistico del campione: conoscere il grado di padronanza della lingua veicolare, funzionale alla comprensione delle consegne, è, senza dubbio, condizione necessaria per una lettura e un'analisi completa dei dati raccolti. Un altro limite potrebbe essere la breve durata dell'intervento di *coding*: 8 incontri svolti in 4 settimane per ogni classe coinvolta nel progetto, analogamente agli studi effettuati in precedenza (Arfé et al., 2019, 2020). Sarebbe interessante che una ricerca specifica indagasse i tempi ottimali dei training per poter evidenziare se il divario di genere, riscontrato nel presente studio, possa essere colmato da tempi di esercizio più dilatati.

Un ruolo fondamentale va attribuito allo spazio per l'apprendimento: i tre laboratori piuttosto obsoleti e non corredati di tutti gli strumenti necessari (es. videoproiettore) non hanno favorito l'agile realizzazione del progetto.

Infine, considerato che l'autoefficacia in maschi e femmine di quarta elementare non sembra differire, potrebbe risultare interessante condurre un test di *follow up* al termine del primo ciclo di istruzione sui bambini, soggetti della presente ricerca, per indagare se l'autoefficacia si modifica con l'aumentare del grado scolastico (Román-González et al., 2017). Ulteriore sviluppo potrebbe essere il confronto con un campione di ragazze e ragazzi coetanei che non hanno mai partecipato ad attività di *coding*.

In conclusione, questo studio ha fornito indicazioni sull'efficacia di un training di *coding* per alunne e alunni di classe quarte della scuola prima, con particolare attenzione alla percezione di autoefficacia legata alla programmazione/*coding*. I risultati hanno mostrato un miglioramento generale nell'accuratezza per entrambi i generi, sebbene siano emerse differenze statisticamente significative nel post-test. In termini di autoefficacia, non sono state riscontrate differenze di genere né a T1 né a T2, suggerendo che la percezione di competenza di bambine e bambini sia simile prima e dopo il l'intervento. Tuttavia, il miglioramento derivante dal training sembra favorire i maschi in termini di autoefficacia totale, mentre le femmine mostrano un vantaggio nella sottoscala *Confidence* la quale

valuta la fiducia degli studenti nelle loro abilità di *coding*. Indagare la presenza di differenze di genere e la percezione di autoefficacia nell'ambito del pensiero computazionale è essenziale per ottenere evidenze più solide rispetto a quelle attualmente presenti in letteratura, dove i risultati sono talvolta contrastanti. Questo potrebbe contribuire a comprendere meglio le radici del divario di genere nelle discipline STEM e a sviluppare strategie innovative per ispirare le bambine e le ragazze a diventare le programmatrici del futuro. Inoltre, al netto di tutti i limiti della presente ricerca si può sostenere che il pensiero computazionale e il *coding* sono strumenti efficaci per l'insegnamento e l'apprendimento della programmazione fin dai primi anni di scuola. Va sottolineato però che tale evidenza si scontra con le criticità del contingente (ambienti per l'apprendimento, formazione dei docenti...) che dovrebbero essere risolte affinché il *Computational Thinking* diventi strutturale al curriculum scolastico.

BIBLIOGRAFIA

- Aho, A. V. (2012). Computation and Computational Thinking. *The Computer Journal*, 55(7), 832–835. <https://doi.org/10.1093/comjnl/bxs074>
- Allaire-Duquette, G., Chastenay, P., Bouffard, T., Bélanger, S. A., Hernandez, O., Mahhou, M. A., ... & Desjarlais, E. (2022). Gender differences in self-efficacy for programming narrowed after a 2-h science museum workshop. *Canadian Journal of Science, Mathematics and Technology Education*, 22(1), 87-100. <https://doi.org/10.1007/s42330-022-00193-7>
- Arfè, B., & Vardanega, T. (2019). Imparare a ragionare: Il ruolo del pensiero computazionale a scuola. *Giornale italiano di psicologia*, 4, 765–768. <https://doi.org/10.1421/95548>
- Arfè, B., Vardanega, T., & Ronconi, L. (2020). The effects of coding on children's planning and inhibition skills. *Computers & Education*, 148, 103807. <https://doi.org/10.1016/j.compedu.2020.103807>
- Arfè, B., Vardanega, T., Montuori, C., & Lavanga, M. (2019). Coding in primary grades boosts children's executive functions. *Frontiers in psychology*, 10, 2713. <https://doi.org/10.3389/fpsyg.2019.02713>
- Baker, D. (2013). What works: Using curriculum and pedagogy to increase girls' interest and participation in science. *Theory Into Practice*, 52(1), 14-20. <https://doi.org/10.1080/07351690.2013.743760>
- Bandura, A. (1986). Social foundations of thought and action. *Englewood Cliffs, NJ*, 1986(23-28), 2.
- Bandura, A. (1997) Editorial. *American Journal of Health Promotion*, 12(1), 8–10.
- Bandura, A. (2000). *Autoefficacia.: Teoria e applicazioni.*(Presentazione all'edizione italiana di Gian Vittorio Caprara). Edizioni Erickson.
- Bandura, A. (2006). Guide for constructing self-efficacy scales. *Self-efficacy beliefs of adolescents*, 5(1), 307-337.
- Bandura, A., & Adams, N. E. (1977). Analysis of self-efficacy theory of behavioral change. *Cognitive therapy and research*, 1(4), 287-310. <https://doi.org/10.1007/BF01663995>
- Bandura, A., & Schunk, D. H. (1981). Cultivating competence, self-efficacy, and intrinsic interest through proximal self-motivation. *Journal of personality and social psychology*, 41(3), 586. <https://doi.org/10.1037/0022-3514.41.3.586>
- Barr, D., Harrison, J., & Conery, L. (2011). Computational thinking: A digital age skill for everyone. *Learning & Leading with Technology*, 38(6), 20-23. <https://eric.ed.gov/?id=EJ918910>
- Beck, A. T. (1979). *Cognitive therapy and the emotional disorders*. Penguin.
- Bers, M. U. (2018). Coding and computational thinking in early childhood: The impact of ScratchJr in Europe. *European Journal of STEM Education*, 3(3), 8. <https://eric.ed.gov/?id=EJ1190774>
- Bourdieu, P. (1990). *The Logic of Practice*. Stanford University Press.

- Brennan, K., & Resnick, M. (2012, April). New frameworks for studying and assessing the development of computational thinking. In *Proceedings of the 2012 annual meeting of the American educational research association, Vancouver, Canada* (Vol. 1, p. 25).
- Brown, I., & Inouye, D. K. (1978). Learned helplessness through modeling: The role of perceived similarity in competence. *Journal of personality and social psychology*, 36(8), 900. <https://doi.org/10.1037/0022-3514.36.8.900>
- Buitrago Flórez, F., Casallas, R., Hernández, M., Reyes, A., Restrepo, S., & Danies, G. (2017). Changing a generation's way of thinking: Teaching computational thinking through programming. *Review of Educational Research*, 87(4), 834-860. <https://doi.org/10.3102/0034654317710096>
- Caprara, G. V. (2001). *La valutazione dell'autoefficacia. Costrutti e strumenti*. Edizioni Erickson.
- Charlesworth, T. E., & Banaji, M. R. (2019). Gender in science, technology, engineering, and mathematics: Issues, causes, solutions. *Journal of Neuroscience*, 39(37), 7228-7243. <https://doi.org/10.1523/JNEUROSCI.0475-18.2019>
- Curzon, P., Black, J., Meagher, L. R., & McOwan, P. (2009). cs4fn. org: Enthusing students about Computer Science. *Proceedings of Informatics Education Europe IV*, 73-80.
- Edgerton, J., Peter, T., & Roberts, L. (2014). Gendered habitus and gender differences in academic achievement. *Alberta journal of educational research*, 60(1), 182-212. <https://doi.org/10.11575/ajer.v60i1.55858>
- Eisenkopf, G., Hessami, Z., Fischbacher, U., & Ursprung, H. W. (2015). Academic performance and single-sex schooling: Evidence from a natural experiment in Switzerland. *Journal of economic behavior & organization*, 115, 123-143. <https://doi.org/10.1016/j.jebo.2014.08.004>
- Espino, E. E. E., & González, C. G. (2016, September). Gender and computational thinking: Review of the literature and applications. In *Proceedings of the XVII International Conference on Human Computer Interaction* (pp. 1-2). <https://doi.org/10.1145/2998626.2998665>
- Francisco, J. M., & Maher, C. A. (2005). Conditions for promoting reasoning in problem solving: Insights from a longitudinal study. *The Journal of Mathematical Behavior*, 24(3-4), 361-372. <https://doi.org/10.1016/j.jmathb.2005.09.001>
- Funke, A., & Geldreich, K. (2017, November). Gender differences in scratch programs of primary school children. In *Proceedings of the 12th workshop on primary and secondary computing education* (pp. 57-64). <https://doi.org/10.1145/3137065.3137067>
- Girelli, L. (2023). What does gender has to do with math? Complex questions require complex answers. *Journal of Neuroscience Research*, 101(5), 679-688. <https://doi.org/10.1002/jnr.25056>
- Grover, S., & Pea, R. (2013). Computational thinking in K-12: A review of the state of the field. *Educational researcher*, 42(1), 38-43. <https://doi.org/10.3102/0013189X12463051>

- Gunbatar, M. S., & Karalar, H. (2018). Gender differences in middle school students' attitudes and self-efficacy perceptions towards mBlock programming. *European Journal of Educational Research*, 7(4), 925-933. <https://doi.org/10.12973/eu-jer.7.4.925>
- Hunt, S. (2012). Computational Thinking. Retrieved from Computing At School
- Kalelioğlu, F. (2015). A new way of teaching programming skills to K-12 students: Code.org. *Computers in Human Behavior*, 52, 200-210. <https://doi.org/10.1016/j.chb.2015.05.047>
- Klein, P. D., Boscolo, P., Gelati, C., & Kirkpatrick, L. C. (2014). New directions in writing as a learning activity. In *Writing as a learning activity* (pp. 1-14). Brill. https://doi.org/10.1163/9789004265011_002
- Lazarus, R. S., & Launier, R. (1978). Stress-related transactions between person and environment. In *Perspectives in interactional psychology* (pp. 287-327). Boston, MA: Springer US. https://link.springer.com/content/pdf/10.1007/978-1-4613-3997-7_12?pdf=chapter%20toc
- Lee, I., Grover, S., Martin, F., Pillai, S., & Malyn-Smith, J. (2020). Computational thinking from a disciplinary perspective: Integrating computational thinking in K-12 science, technology, engineering, and mathematics education. *Journal of Science Education and Technology*, 29(1), 1-8. <https://doi.org/10.1007/s10956-019-09803-w>
- Lewis, C. M., & Shah, N. (2012, February). Building upon and enriching grade four mathematics standards with programming curriculum. In *Proceedings of the 43rd ACM technical symposium on Computer Science Education* (pp. 57-62). <https://doi.org/10.1145/2157136.215715>
- Li, Y., Schoenfeld, A. H., diSessa, A. A., Graesser, A. C., Benson, L. C., English, L. D., & Duschl, R. A. (2020). On computational thinking and STEM education. *Journal for STEM Education Research*, 3(2), 147-166. <https://doi.org/10.1007/s41979-020-00044-w>
- Maloney, E. A., Waechter, S., Risko, E. F., & Fugelsang, J. A. (2012). Reducing the sex difference in math anxiety: The role of spatial processing ability. *Learning and Individual Differences*, 22(3), 380-384. <https://doi.org/10.1016/j.lindif.2012.01.001>
- Martin, F. (2018). Rethinking computational thinking. Retrieved October 24, 2020 from <http://advocate.csteachers.org/2018/02/17/rethinking-computational-thinking/>.
- Mason, L. (2001). Introducing talk and writing for conceptual change: A classroom study. *Learning and instruction*, 11(4-5), 305-329. [https://doi.org/10.1016/S0959-4752\(00\)00035-9](https://doi.org/10.1016/S0959-4752(00)00035-9)
- Mason, S. L., & Rich, P. J. (2020). Development and analysis of the elementary student coding attitudes survey. *Computers & Education*, 153, 103898. <https://doi.org/10.1016/j.compedu.2020.103898>
- Master, A., Cheryan, S., Moscatelli, A., & Meltzoff, A. N. (2017). Programming experience promotes higher STEM motivation among first-grade girls. *Journal of experimental child psychology*, 160, 92-106. <https://doi.org/10.1016/j.jecp.2017.03.013>
- Meichenbaum, D. H. *Cognitive-behavior modification: An integrative approach*. New York: Plenum Press, 1977.

- Metz, S. S. (2007). Attracting the engineers of 2020 today. *Women and minorities in science, technology, engineering, and mathematics: Upping the numbers*, 184-209.
- Miyake, A., Kost-Smith, L. E., Finkelstein, N. D., Pollock, S. J., Cohen, G. L., & Ito, T. A. (2010). Reducing the gender achievement gap in college science: A classroom study of values affirmation. *Science*, 330(6008), 1234-1237. <https://doi.org/10.1126/science.1195996>
- Mohaghegh, D. M., & McCauley, M. (2016). Computational thinking: The skill set of the 21st century.
- Montuori, C., Ronconi, L., Vardanega, T., & Arfé, B. (2022). Exploring gender differences in coding at the beginning of primary school. *Frontiers in Psychology*, 13, 887280. <https://doi.org/10.3389/fpsyg.2022.887280>
- Nardelli, E., & Ventre, G. (2015). Introducing computational thinking in Italian schools: A first report on “programma il futuro” project. In *INTED2015 Proceedings* (pp. 7414-7421). IATED. <https://library.iated.org/view/NARDELLI2015INT>
- Orton, K., Weintrop, D., Beheshti, E., Horn, M., Jona, K., & Wilensky, U. (2016). Bringing computational thinking into high school mathematics and science classrooms. Singapore: International Society of the Learning Sciences. <https://repository.isls.org/handle/1/183>
- Papavlasopoulou, S., Sharma, K., & Giannakos, M. N. (2020). Coding activities for children: Coupling eye-tracking with qualitative data to investigate gender differences. *Computers in Human Behavior*, 105, 105939. <https://doi.org/10.1016/j.chb.2019.03.003>
- Papert, S. (1980). *Mindstorms: Computers, children, and powerful ideas*. NY: Basic Books, 255, 15.
- Papert, S. (1996). An exploration in the space of mathematics educations. *Int. J. Comput. Math. Learn.*, 1(1), 95-123.
- Phillips, P. (2007). Computational thinking: A problem-solving tool for
- Ramalingam, V., & Wiedenbeck, S. (1998). Development and validation of scores on a computer programming self-efficacy scale and group analyses of novice programmer self-efficacy. *Journal of Educational Computing Research*, 19(4), 367-381. <https://doi.org/10.2190/C670-Y3C8-LTJ1-CT3P>
- Repenning, A., Webb, D., & Ioannidou, A. (2010, March). Scalable game design and the development of a checklist for getting computational thinking into public schools. In *Proceedings of the 41st ACM technical symposium on Computer science education* (pp. 265-269). <https://doi.org/10.1145/1734263.1734357>
- Román-González, M., Pérez-González, J. C., & Jiménez-Fernández, C. (2017). Which cognitive abilities underlie computational thinking? Criterion validity of the Computational Thinking Test. *Computers in human behavior*, 72, 678-691. <https://doi.org/10.1016/j.chb.2016.08.047>
- Sáez-López, J. M., Román-González, M., & Vázquez-Cano, E. (2016). Visual programming languages integrated across the curriculum in elementary school: A two year case study using “Scratch” in five schools. *Computers & Education*, 97, 129-141. <https://doi.org/10.1016/j.compedu.2016.03.003>

- Sarason, I. G. Anxiety and self-preoccupation. In I. G. Sarason & D. C. Spielberger (Eds.), *Stress and anxiety* (Vol.2). Washington, D.C.: Hemisphere, 1975.
- Schunk, D. H. (1981). Modeling and attributional effects on children's achievement: A self-efficacy analysis. *Journal of educational psychology*, 73(1), 93. <https://doi.org/10.1037/0022-0663.73.1.93>
- Sengupta, P., Kinnebrew, J. S., Basu, S., Biswas, G., & Clark, D. (2013). Integrating computational thinking with K-12 science education using agent-based computation: A theoretical framework. *Education and Information Technologies*, 18, 351-380. <https://doi.org/10.1007/s10639-012-9240-x>
- Shute, V. J., Sun, C., & Asbell-Clarke, J. (2017). Demystifying computational thinking. *Educational research review*, 22, 142-158. <https://doi.org/10.1016/j.edurev.2017.09.003>
- Steele, C. M. (1997). A threat in the air: How stereotypes shape intellectual identity and performance. *American psychologist*, 52(6), 613.
- Strawhacker, A., & Bers, M. U. (2019). What they learn when they learn coding: Investigating cognitive domains and computer programming knowledge in young children. *Educational technology research and development*, 67, 541-575. <https://doi.org/10.1007/s11423-018-9622-x>
- Sullivan, A., & Bers, M. U. (2016). Girls, boys, and bots: Gender differences in young children's performance on robotics and programming tasks. *Journal of Information Technology Education. Innovations in Practice*, 15, 145.
- Tellhed, U., Björklund, F., & Strand, K. K. (2022). Sure I can code (but do I want to?). Why boys' and girls' programming beliefs differ and the effects of mandatory programming education. *Computers in Human Behavior*, 135, 107370. <https://doi.org/10.1016/j.chb.2022.107370>
- Tsarava, K., Moeller, K., Román-González, M., Golle, J., Leifheit, L., Butz, M. V., & Ninaus, M. (2022). A cognitive definition of computational thinking in primary education. *Computers & Education*, 179, 104425. <https://doi.org/10.1016/j.compedu.2021.104425>
- Turnbull, S. M., Locke, K., Vanholsbeeck, F., & O'Neale, D. R. (2019). Bourdieu, networks, and movements: Using the concepts of habitus, field and capital to understand a network analysis of gender differences in undergraduate physics. *PloS one*, 14(9), e0222357. <https://doi.org/10.1371/journal.pone.0222357>
- UNESCO. (2017). *Education for Sustainable Development Goals: Learning Objectives*. UNESCO. <https://unesdoc.unesco.org/ark:/48223/pf0000253479>
- Van den Broek, P. (1989). Causal reasoning and inference making in judging the importance of story statements. *Child development*, 286-297. <https://doi.org/10.2307/1130976>
- Wang, C., Shen, J., & Chao, J. (2022). Integrating computational thinking in STEM education: A literature review. *International Journal of Science and Mathematics Education*, 20(8), 1949-1972. <https://doi.org/10.1007/s10763-021-10227-5>

- Wang, M. T., & Degol, J. L. (2017). Gender gap in science, technology, engineering, and mathematics (STEM): Current knowledge, implications for practice, policy, and future directions. *Educational psychology review*, 29, 119-140. <https://doi.org/10.1007/s10648-015-9355-x>
- Weinberg, R., Gould, D., & Jackson, A. (1979). Expectations and performance: An empirical test of Bandura's self-efficacy theory. *Journal of Sport and Exercise Psychology*, 1(4), 320-331. <https://doi.org/10.1123/jsp.1.4.320>
- Weintrop, D., Beheshti, E., Horn, M. S., Orton, K., Trouille, L., Jona, K., et al. (2014). Interactive assessment tools for computational thinking in high school STEM classrooms. *Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering*, 136. https://doi.org/10.1007/978-3-319-08189-2_3
- Wilensky, U., & Reisman, K. (2006). Thinking like a wolf, a sheep, or a firefly: Learning biology through constructing and testing computational theories—an embodied modeling approach. *Cognition and instruction*, 24(2), 171-209. https://doi.org/10.1207/s1532690xci2402_1
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35. <https://doi.org/10.1145/1118178.1118215>
- Wing, J. M. (2010). Computational Thinking What and Why.

SITOGRAFIA

Code.org. *Learn computer science. Change the world.* <https://code.org>

Gazzetta Ufficiale. (2015, July 15). *Gazzetta ufficiale della repubblica italiana.* <https://www.gazzettaufficiale.it/eli/id/2015/07/15/15G00122/sg>

Istat. (2023, October). *Report livelli di istruzione e ritorni occupazionali.* <https://www.istat.it/it/files/2023/10/Report-livelli-di-istruzione-e-ritorni-occupazionali.pdf>

Ministero dell'Istruzione e del Merito. (2023, October 24) *Linee guida STEM.* <https://www.miur.gov.it/documents/20182/0/Linee+guida+STEM.pdf/2aa0b11f-7609-66ac-3fd8-2c6a03c80f77?version=1.0&t=1698173043586>

Programma il Futuro. (n.d). *Programma il Futuro - Code.org - ProgrammailFuturo.it.* <https://programmailfuturo.it>

UNESCO. (2017). *Education for Sustainable Development Goals: Learning Objectives.* <https://unesdoc.unesco.org/ark:/48223/pf0000253479>