



**UNIVERSITÀ DEGLI STUDI DI PADOVA**

**Dipartimento di Psicologia Generale**

**Corso di laurea Magistrale in Psicologia Cognitiva Applicata**

**Tesi di laurea Magistrale**

**DIFFERENZE DI GENERE E AUTOEFFICACIA NEL  
CODING NELLA SCUOLA PRIMARIA**

**Gender Gap and Coding Self-Efficacy in Primary School**

*Relatrice*

**Prof.ssa** Barbara Arfè

*Correlatrice*

**Dott.ssa** Costanza Padova

*Laureanda:* Simona Russo

*Matricola:* 2055504

## Indice

<b>Introduzione</b> .....	<b>1</b>
<b>Capitolo 1. Il Pensiero Computazionale</b> .....	<b>4</b>
1.1 Basi sul Pensiero Computazionale.....	4
1.1.1 Processi sottostanti al Pensiero Computazionale .....	5
1.1.2 Il Pensiero Computazionale in ambito educativo.....	6
1.2 Pensiero Computazionale e <i>coding</i> .....	9
1.2.1 Effetto del <i>coding</i> sulle Funzioni Esecutive.....	10
1.3 Training del Pensiero Computazionale .....	12
1.3.1 Coding <i>Plugged</i> .....	13
1.3.2 Coding <i>Unplugged</i> .....	15
<b>Capitolo 2. Differenze di genere nelle discipline STEM e nel <i>coding</i></b> .....	<b>18</b>
2.1 STEM e differenze di genere .....	18
2.2 <i>Coding</i> e differenze di genere .....	21
<b>Capitolo 3. L'autoefficacia</b> .....	<b>24</b>
3.1 Basi sull'autoefficacia.....	24
3.1.1 Autoefficacia e concetti affini .....	25
3.1.2 Sostenere l'autoefficacia .....	26
3.1.3 Autoefficacia scolastica.....	27
3.2 Autoefficacia legata al <i>coding</i> .....	29
3.2.1 Differenze di genere nell'autoefficacia legata al <i>coding</i> .....	31
<b>Capitolo 4. Metodo</b> .....	<b>35</b>
4.1 Obiettivo di ricerca .....	35
4.1.1 Disegno sperimentale .....	36
4.2 Partecipanti .....	38
4.3 Strumenti.....	40
4.3.1 Funzioni Esecutive .....	40
4.3.2 Misure di autoefficacia.....	43
4.3.3 Questionario sulle teorie dell'intelligenza.....	45
4.3.4 Abilità di <i>coding</i> .....	46
4.4 Procedura .....	53
<b>Capitolo 5. Risultati</b> .....	<b>56</b>
5.1 Analisi dei dati .....	56
5.1.1 Analisi preliminari.....	56

5.1.2 Valutazione dell'intervento di training sulle abilità di <i>coding</i> .....	58
5.1.3 Differenze di genere nel miglioramento delle abilità di <i>coding</i> .....	61
5.1.4 Relazione tra autoefficacia e abilità di <i>coding</i> .....	62
5.2 Discussione e conclusioni .....	68
5.2.1 Limiti e sviluppi futuri .....	71
<b>Bibliografia</b> .....	<b>73</b>
<b>Sitografia</b> .....	<b>82</b>

## Introduzione

Uno degli obiettivi del Digital Education Action Plan 2021-2027 dell'Unione Europea (Bocconi et al., 2022) è l'introduzione nelle scuole dell'obbligo di concetti base di informatica già durante l'infanzia. La pratica dell'informatica rende possibile lo sviluppo del *Computational Thinking*, o pensiero computazionale, strumento utile per migliorare le abilità di problem solving e applicabile in ogni ambito e disciplina, inclusa la vita quotidiana. Inoltre, introdurre l'informatica a bambini in tenera età potrebbe contribuire a ridurre il divario di genere ampiamente documentato in ambito STEM (l'insieme delle discipline scientifico-tecnologiche), in cui le donne risultano sottorappresentate in termini occupazionali e di istruzione (UNESCO, 2017).

In letteratura sono presenti evidenze che hanno mostrato l'efficacia di un training di *coding* sullo sviluppo delle abilità del pensiero computazionale, potenziando inoltre le Funzioni Esecutive in alunni di classi prime della scuola primaria (Arfé et al., 2019, 2020; Montuori et al., 2022). Lo studio di Montuori et al., (2022) ha inoltre riscontrato la presenza di differenze di genere sull'effetto di miglioramento delle abilità di *coding*, risultate maggiori per i maschi a seguito dell'intervento di training. Tuttavia, in letteratura le evidenze sulle differenze di genere nel *coding* sono contrastanti, sia rispetto alla presenza che alla direzione, cioè se a vantaggio dei maschi o delle femmine.

Alcuni studi hanno mostrato che le abilità di *coding* possono essere influenzate dall'autoefficacia, ovvero le convinzioni personali riguardo alle capacità di riuscita di un compito. Tuttavia, date le evidenze limitate e contrastanti, non risulta chiaro se le possibili differenze di genere nelle abilità di *coding* possano essere influenzate da differenze di genere nel senso di autoefficacia, nello specifico l'autoefficacia STEM o legata al *coding*.

Inoltre, gli studi che indagano questo effetto sono stati principalmente svolti con alunni di scuola secondaria di primo grado o di età maggiore.

Questo elaborato si pone come obiettivo quello di mostrare i risultati di uno studio volto a indagare la presenza di una differenza di genere nelle abilità di *coding* in un campione alunni di classi quarte della scuola primaria, esaminando il ruolo dell'autoefficacia in questo effetto.

Lo studio si sviluppa all'interno del progetto *Computational Thinking* (CoThi), promosso dal Dipartimento di Psicologia dello Sviluppo e della Socializzazione dell'Università degli Studi di Padova, in collaborazione con il Dipartimento di Matematica.

I primi tre capitoli hanno lo scopo di fornire un quadro teorico sul pensiero computazionale, l'autoefficacia e la presenza di differenze di genere in questo ambito.

Nel primo capitolo verrà data una definizione di *Computational Thinking*, mostrando i processi cognitivi alla base del suo sviluppo. Oltre a fornire una panoramica sulla sua integrazione in ambito educativo in diversi Stati Europei, verrà descritto, sulla base delle evidenze presenti in letteratura, uno strumento risultato efficace per lo sviluppo delle abilità del pensiero computazionale: il *coding*.

Il secondo capitolo tratterà delle differenze di genere in ambito STEM, descrivendo possibili spiegazioni legate a fattori biologici, sociali e culturali, con il supporto di studi in merito. In questo capitolo verrà inoltre mostrata la letteratura che indaga la presenza di differenze di genere nelle abilità di *coding*.

Nel terzo capitolo verrà fornita una definizione del concetto di autoefficacia a partire dalle formulazioni teoriche dello psicologo Albert Bandura, descrivendo i fattori che la influenzano e le strategie per incrementarla. Verrà poi introdotto il concetto di autoefficacia scolastica, rilevante in questo contesto dato il campione di riferimento di

studenti di scuola primaria. Infine, verranno riportate alcune evidenze sulla relazione tra l'autoefficacia e le abilità di *coding*, nonché alcuni studi che indagano la presenza di differenze di genere in questa relazione. Tuttavia, la letteratura in questo specifico ambito è limitata e non sempre le evidenze riscontrate sono in accordo tra loro. Per questo motivo risulta importante contribuire a fare ricerca su questo fenomeno, con lo scopo di giungere a risultati più solidi.

I due capitoli successivi descriveranno invece in modo approfondito lo studio condotto.

Prima di tutto verrà spiegata la metodologia di ricerca, illustrando dettagliatamente l'obiettivo di ricerca, le ipotesi testate e il disegno sperimentale. Verranno presentati gli strumenti utilizzati per valutare sia il livello di autoefficacia che le abilità di *coding*, oltre che i test di valutazione delle Funzioni Esecutive. Inoltre, verrà descritta la procedura di intervento.

Nel capitolo conclusivo verranno spiegate le modalità di analisi statistica dei dati e saranno illustrati i risultati ottenuti in merito alla valutazione dell'intervento, alla presenza di differenze di genere sull'effetto del trattamento e alla relazione tra l'autoefficacia e le abilità di *coding*. Infine, verranno descritti i limiti dello studio e forniti spunti per possibili sviluppi futuri.

# Capitolo 1. Il Pensiero Computazionale

## 1.1 Basi sul Pensiero Computazionale

L'espressione "*Computational Thinking*" (CT) viene usata per la prima volta da Seymour Papert nel suo libro *Mindstorms*, pubblicato nel 1980. Sebbene non ci sia totale accordo sulla sua definizione (Román-González et al., 2017), quando si parla di CT, o pensiero computazionale, il principale punto di riferimento è l'articolo pubblicato nel 2006 da Wing, capo del Dipartimento di Computer Science della Carnegie Mellon University, che per prima ne ha dato una definizione accurata.

Wing (2006) descrive il CT come un'abilità fondamentale per chiunque, non solo gli informatici, che riguarda la risoluzione di problemi, la progettazione e la comprensione del comportamento umano, attingendo dai concetti base dell'informatica. Nel 2010, Wing approfondisce maggiormente la definizione di *Computational Thinking*, descrivendolo come l'insieme dei processi di pensiero coinvolti nella formulazione di problemi e delle relative soluzioni, rappresentate in modo tale che possano essere efficientemente eseguite da un agente di elaborazione delle informazioni. Aho (2012) riprende questa definizione, specificando che le soluzioni debbano essere rappresentate sotto forma di step computazionali e algoritmi.

Wing (2010) sottolinea che sviluppare il pensiero computazionale non significa imparare a programmare, né imparare a pensare come un computer, bensì utilizzare un approccio alla risoluzione di problemi, applicabile in ogni ambito e disciplina, non solo in matematica, informatica o ingegneria, ma anche in discipline quali medicina, archeologia, economia, scienze sociali ecc., così come nella vita quotidiana. Il computer diventa quindi solamente un mezzo di semplificazione e supporto alla formulazione e risoluzione di problemi.

Tikva e Tambouris (2021), nella loro recente review sistematica della letteratura riguardante il pensiero computazionale in ambito educativo, esaminano le definizioni di *Computational Thinking*, dividendole in due categorie: la prima dominio-specifica, inerente al problem solving in ambito informatico; la seconda di tipo dominio-generale che riguarda il problem solving nella vita quotidiana, in accordo con quanto affermato da Wing (2010).

### 1.1.1 Processi sottostanti al Pensiero Computazionale

Arfè e Vardanega (2019) affermano che “il pensiero computazionale riguarda una forma specifica di ragionamento: il ragionamento attorno ai problemi” (p. 766). Il tipo di ragionamento che sta alla base del pensiero computazionale non è quello intuitivo, piuttosto quello analitico e lento. In particolare, il CT coinvolge i processi di pensiero deduttivo, induttivo, abduttivo e analogico (Arfè & Vardanega, 2019; Bucciarelli, 2019).

Alla base del CT c'è la capacità di astrazione, un processo induttivo che permette di semplificare e categorizzare i problemi, definire pattern generalizzabili e costruire strategie di risoluzione efficienti (Aho, 2012; Wing, 2010; Yasar, 2018). Un altro importante processo coinvolto, di tipo deduttivo, è la decomposizione che consiste nel suddividere un problema in parti più piccole e più semplici, la cui risoluzione, pezzo per pezzo, porta ad una soluzione cumulativa del problema iniziale (Arfè & Vardanega, 2019; Yasar, 2018).

Altri processi coinvolti sono:

- **Generalizzazione:** individuare le somiglianze tra problemi, identificando e applicando uno stesso pattern di risoluzione a una vasta gamma di problemi (Arfè & Vardanega, 2019; Román-González et al., 2017);
- **Pensiero algoritmico:** processo che coinvolge il ragionamento di tipo abduttivo, dagli effetti alle cause, e consiste nel rappresentare i problemi attraverso una serie



di step ordinati logicamente, che rendono possibile automatizzare le soluzioni attraverso l'utilizzo di computer (Bucciarelli, 2019; Román-González et al., 2017);

- **Verifica logica:** processo di tipo analitico che consiste nella valutazione della strategia risolutiva individuata, con lo scopo di trovare e correggere gli errori (*debugging*) per arrivare a procedure più efficienti o sintetiche (Arfé & Vardanega, 2019; Montuori et al., 2022).

A livello concreto, sviluppare il pensiero computazionale permette di acquisire le seguenti abilità, utili per scienziati, ingegneri e altri professionisti (Wing, 2010):

- Applicare nuovi metodi computazionali ai loro problemi;
- Riformulare i problemi in modo che siano adatti alla computazione;
- Spiegare problemi e soluzioni in termini computazionali;
- Analizzare dati di grandi dimensioni, con la possibilità di esplorare nuove domande di ricerca e scoprire nuove discipline e ambiti scientifici.

Le abilità acquisite utili per chiunque sono invece (Wing, 2010):

- Capire quali aspetti di un problema sono adatti alla computazione;
- Valutare i limiti e i punti di forza degli strumenti e delle tecniche computazionali e capire la loro corrispondenza con un problema;
- Applicare o adattare uno strumento o tecnica computazionale a un nuovo utilizzo o in nuovi domini.

### **1.1.2 Il Pensiero Computazionale in ambito educativo**

Un concetto su cui vale la pena porre l'attenzione è quello espresso da Nardelli (2019). L'autore sottolinea che non si deve pensare al *Computational Thinking* come ad una nuova disciplina da insegnare, ma come a un set di competenze mentali e cognitive che si apprendono grazie allo studio e alla pratica dell'informatica. La relazione tra

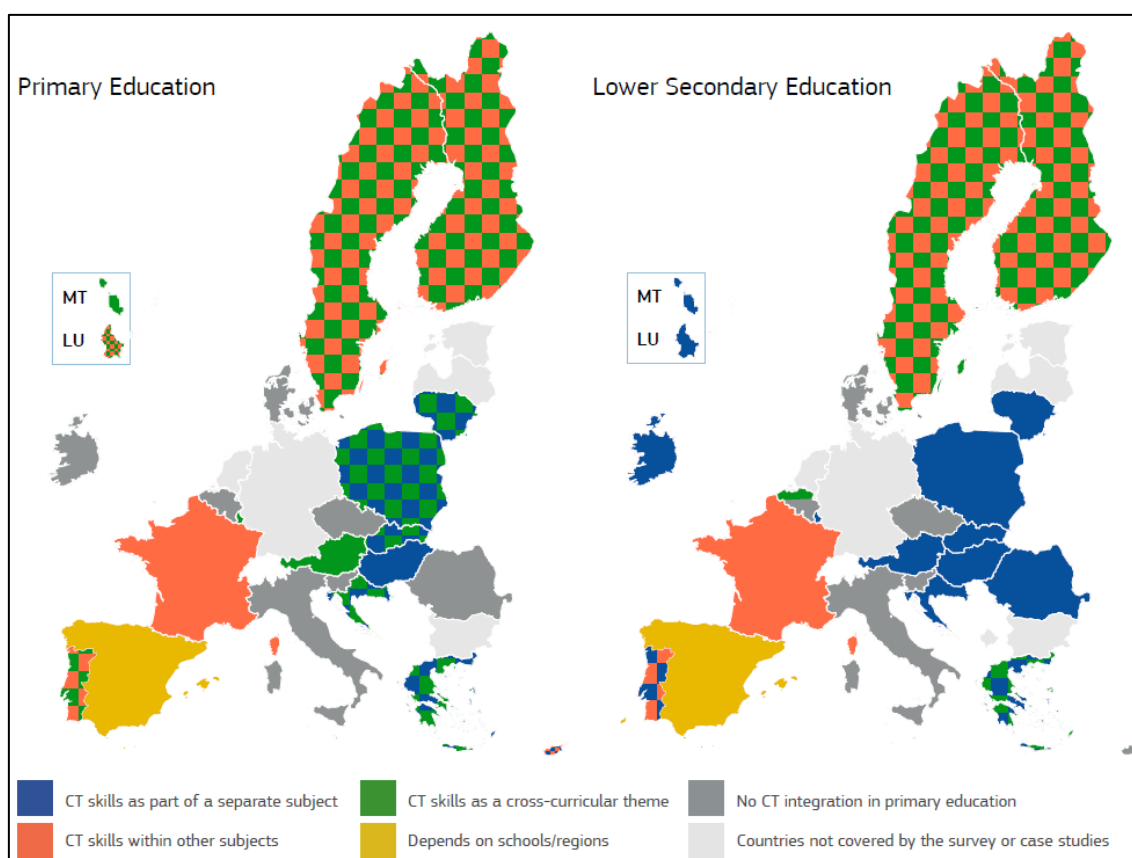
l'informatica e il pensiero computazionale è la stessa che sussiste tra la matematica e il pensiero di tipo logico-matematico: da una parte c'è la disciplina che viene insegnata, dall'altra i principi e il nocciolo concettuale di tale disciplina e le competenze che vengono apprese tramite il suo apprendimento (Nardelli, 2019).

Per questo motivo, il focus dovrebbe essere non tanto su come insegnare il CT, ma su come insegnare e valutare le competenze informatiche apprese durante gli anni di scolarizzazione (Nardelli, 2019). Sempre più paesi, come Regno Unito, Stati Uniti e Francia, stanno inserendo l'informatica nei curriculum di giovani studenti, già a partire dalla scuola primaria e in alcuni casi, anche durante la scuola dell'infanzia (Nardelli, 2019). Iniziare a esporre i bambini ai concetti dell'informatica in tenera età sta diventando sempre più rilevante ed è uno degli obiettivi del Digital Education Action Plan 2021-2027 dell'Unione Europea (Bocconi et al., 2022). L'obiettivo risulta in linea con l'opinione di molti studiosi che considerano la computazione come il terzo pilastro del metodo scientifico, insieme alla teoria e alla sperimentazione (Riedel et al., 2008; Wing, 2010). Questo non perché il modo di pensare degli informatici sia migliore rispetto agli altri, ma perché offre un paradigma concettuale complementare e utile per descrivere la realtà (Nardelli, 2019).

Nel 2022 l'Unione Europea, tramite il Joint Research Centre, ha pubblicato un report riguardo all'integrazione del pensiero computazionale nelle scuole dell'obbligo di 29 Paesi Europei, di cui 22 Stati Membri dell'UE, analizzandone i curricula tra il 2016 e il 2021 (Bocconi et al., 2022). Secondo il report, 18 degli Stati Membri dell'UE e sette Paesi extra UE hanno già integrato obbligatoriamente l'introduzione di concetti base di informatica nei loro curricula, con lo scopo di sviluppare le abilità del pensiero computazionale. Tra i quattro paesi rimanenti, la Danimarca ha messo in atto un'iniziativa pilota, mentre Repubblica Ceca, Italia e Slovenia hanno in programma politiche in questa

direzione, ponendo l'avviamento al pensiero computazionale come obiettivo, piuttosto che come un obbligo formativo.

Il report ha anche indagato in che modo il CT è stato integrato, individuando tre modalità principali (**Figura 1**): 1) come disciplina autonoma; 2) all'interno di altre materie specifiche quali matematica o tecnologia; 3) come insegnamento transdisciplinare.



**Figura 1.** Modalità di integrazione del *Computational Thinking* nei paesi aderenti all'indagine, separatamente per scuola primaria e scuola secondaria di primo grado (MT = Malta; LU = Lussemburgo) (Bocconi et al., 2022).

Anche altri Paesi Europei non presi in analisi nel report, nello specifico 12 Stati Membri dell'UE e cinque Paesi non UE, hanno introdotto obbligatoriamente i concetti base dell'informatica nei programmi di scuola primaria e scuola secondaria di primo grado.

In Italia, l'introduzione del pensiero computazionale nelle scuole è avvenuta nel 2014, attraverso l'iniziativa "Programma il Futuro", avviata dal Ministero dell'Istruzione e del Merito (MIM, allora MIUR), in collaborazione con il CINI (Consorzio Interuniversitario Nazionale per l'Informatica) (Nardelli & Ventre, 2015). Lo scopo del progetto, tutt'ora in corso, è fornire alle scuole uno strumento per l'insegnamento dei concetti base dell'informatica attraverso attività ludiche. Un anno dopo rispetto all'introduzione dell'iniziativa, la legge 107/2015 riporta il pensiero computazionale tra gli obiettivi di istruzione per le competenze digitali (Arfè & Vardanega, 2019). È solo nel 2018 che il documento "Indicazioni nazionali e nuovi scenari", redatto dal Comitato Scientifico Nazionale istituito dal MIUR, equipara l'avviamento al pensiero computazionale a discipline quali matematica, storia, lingue e geografia e lo promuove come una nuova disciplina nella scuola dell'infanzia e dell'obbligo, il cui scopo non è tanto insegnare un linguaggio di programmazione, ma insegnare una forma di ragionamento per arrivare a soluzioni computazionali (Arfè & Vardanega, 2019; Bocconi et al., 2022).

## **1.2 Pensiero Computazionale e *coding***

Se il *Computational Thinking* è un approccio alla risoluzione di problemi, il *coding* è lo strumento con cui concretamente mettere in pratica le abilità del pensiero computazionale. Consiste nel generare istruzioni sottoforma di codice, *step by step*, così da progettare un piano di esecuzione, i cui effetti sul problema possono essere verificati empiricamente (Montuori et al., 2022). Le istruzioni generate permettono quindi a persone o macchine di mettere in atto una o più azioni.

### 1.2.1 Effetto del *coding* sulle Funzioni Esecutive

Le funzioni esecutive (FE) sono l'insieme di abilità di ordine superiore che supportano la regolazione cognitiva e risultano fondamentali nei compiti di pianificazione e problem solving (Miyake et al., 2000). Le tre componenti principali delle FE sono:

1. **Flessibilità:** capacità di spostare l'attenzione tra compiti diversi, che richiedono differenti *mental set* (Miyake et al., 2000; Nesbitt et al., 2013);
2. **Memoria di lavoro:** capacità di trattenere le informazioni sensoriali, rilevanti per lo svolgimento di un determinato compito (Baddeley, 2010);
3. **Inibizione:** capacità di controllare e inibire deliberatamente le risposte di tipo impulsivo, dominanti o automatiche quando necessario (Miyake et al., 2000).

Il potenziamento delle FE, soprattutto in età scolare, risulta fondamentale per un corretto processo di sviluppo. Infatti, durante il periodo prescolare e nei primi anni di scuola primaria, tra i 5 e i 7 anni circa, i bambini vanno incontro a rapidi cambiamenti del funzionamento cognitivo, con effetti a lungo termine, influenzando i futuri risultati accademici e le abilità di regolazione (Arfé et al., 2019; Di Lieto et al., 2017).

Le tre componenti delle funzioni esecutive risultano coinvolte nelle attività di sviluppo del pensiero computazionale, in particolare la memoria di lavoro (Di Lieto et al., 2017; Shute et al., 2017) e l'inibizione (Arfé et al., 2019, 2020; Di Lieto et al., 2017), così come la pianificazione (Arfé et al., 2019, 2020).

La letteratura sugli effetti del *coding* sullo sviluppo delle FE risulta limitata. Di Lieto e colleghi (2017) hanno svolto uno studio pilota con 12 bambini in età prescolare (5-6 anni) con lo scopo di valutare l'effetto a breve termine di un training di *coding*, nello specifico di Educazione Robotica, sulle FE. Nel corso di sei settimane sono state svolte 13 sessioni di training tramite Bee-Bot, un robot a forma di ape che richiede di essere programmato, cliccando le frecce direzionali in base al percorso da seguire. I bambini

sono stati testati in tre diversi tempi con una batteria neuropsicologica: baseline, pre-training (dopo cinque settimane dal baseline, in cui sono state portate avanti le regolari attività didattiche) e post-training. I risultati hanno mostrato un miglioramento significativo della memoria di lavoro visuo-spaziale e delle capacità di inibizione a seguito del training, oltre che migliori capacità di *coding*. Questo suggerisce che sviluppare il pensiero computazionale favorisce lo sviluppo delle funzioni esecutive.

Nella loro pubblicazione del 2019, Arfè et al. hanno descritto due studi svolti per testare gli effetti del *coding* sullo sviluppo delle FE, ipotizzando un effetto benefico sulle abilità di pianificazione e inibizione delle risposte in alunni della scuola primaria. La fascia d'età considerata era simile a quella dello studio appena descritto (Di Lieto et al., 2017). In particolare, sono stati coinvolti 76 alunni di classe prima per il primo studio (5-6 anni) e 36 alunni di classe seconda per il secondo studio (6-7 anni). Nel primo studio, di tipo randomizzato e controllato, sono stati confrontati gli effetti sulle abilità di pianificazione e inibizione dovuti a un mese di attività di *coding* tramite la piattaforma Code.org con quelli dovuti ad attività STEM standard. I bambini sono stati testati in tre diversi tempi: prima di iniziare le attività (T1); a distanza di un mese (T2), durante il quale il gruppo sperimentale ha svolto il training di *coding*, mentre il gruppo di controllo ha svolto attività STEM; un mese dopo il T2, a seguito delle attività di training di *coding* per il gruppo di controllo e delle attività STEM per il gruppo sperimentale (T3). Il secondo studio, oltre che testare gli effetti di un mese di attività di *coding* contro attività STEM in due gruppi diversi, così come il primo studio, ha anche valutato longitudinalmente il gruppo sperimentale. Gli alunni appartenenti al gruppo sperimentale sono stati testati sette mesi prima dell'inizio delle attività di training, durante il quale hanno svolto le regolari attività didattiche. Lo scopo era verificare se i miglioramenti nelle abilità di pianificazione e inibizione a seguito del training fossero maggiori rispetto a quelli ottenuti dagli stessi

bambini in sette mesi di fisiologico sviluppo con attività curriculari standard. I risultati di entrambi gli studi hanno mostrato i benefici del *coding* sulle funzioni esecutive, con performance significativamente migliori nei compiti di pianificazione e inibizione della risposta del gruppo sperimentale rispetto al gruppo di controllo. Inoltre, i dati longitudinali hanno mostrato che i miglioramenti ottenuti con il training erano equivalenti o superiori rispetto al miglioramento ottenuto con sette mesi di attività standard. Questi risultati supportano l'ipotesi che l'utilizzo di un ambiente virtuale, come Code.org, può supportare lo sviluppo delle FE. Un'ulteriore prova di questo sono i risultati di uno studio successivo (Arfé et al., 2020), svolto con un campione di 179 alunni di classe prima della scuola primaria. In particolare, è stato riscontrato che il training attraverso Code.org porta sia ad un miglioramento significativo nel risolvere problemi di *coding*, sia a migliori abilità di inibizione delle risposte e di risoluzione di compiti standardizzati di pianificazione, con un miglioramento significativo dell'accuratezza e un aumento del tempo trascorso a pianificare.

Date queste evidenze, risulta importante introdurre il pensiero computazionale nei primi anni di scuola, per sfruttare al meglio la finestra di massimo sviluppo delle FE (Arfé et al., 2019; Di Lieto et al., 2017).

### **1.3 Training del Pensiero Computazionale**

Il *coding* si è dimostrato essere un valido strumento per allenare il pensiero computazionale (Arfé et al., 2019; Flórez et al., 2017; Sáez-López et al., 2016). Come precedentemente discusso, le abilità del CT non richiedono necessariamente l'utilizzo di un computer, ma si possono applicare anche nella vita quotidiana (Wing, 2006, 2010). Allo stesso modo, l'insegnamento del *coding* può avvenire sia tramite computer (*coding plugged*) che senza il supporto di strumenti digitali, ad esempio con carta e penna (*coding unplugged*) (Arfé et al., 2019). In entrambi i casi, l'obiettivo è 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 (Arfé et al., 2019).

### **1.3.1 Coding Plugged**

Alcune piattaforme utilizzate per insegnare il *coding*, come Code.org e Scratch, sono basate sulla programmazione visiva a blocchi, modalità che risulta molto efficace per supportare lo sviluppo del CT (Tawfik et al., 2022), anche con bambini ai primi anni di scuola (Arfé et al., 2019; Kalelioğlu, 2015; Sáez-López et al., 2016).

Code.org è una piattaforma di programmazione open-source, che permette a un'ampia gamma di utenti, tra cui giovani studenti e istituti scolastici, di avere accesso all'informatica (Kalelioğlu, 2015; Nardelli & Ventre, 2015). La piattaforma è stata lanciata nel 2013 negli USA dall'omonima organizzazione no-profit tramite il progetto "Hour of Code" (L'Ora del Codice), ponendosi come obiettivo a breve termine quello di dare l'opportunità a tutti gli studenti in tutto il mondo di fare almeno un'ora di programmazione tramite la piattaforma, di facile accessibilità, mentre come obiettivo a lungo termine quello di fornire ad ogni studente un adeguato insegnamento in ambito informatico (Nardelli & Ventre, 2015). In Italia, il progetto è stato introdotto nel 2014 tramite l'iniziativa "Programma il Futuro", implementando un sito a supporto degli insegnanti (<http://programmailfuturo.it>) (Nardelli & Ventre, 2015). Il sito contiene molti tutorial, divisi per fascia d'età e difficoltà, ognuno formato dai 10 ai 20 esercizi, la cui difficoltà aumenta gradualmente. I tutorial sono molto intuitivi e, con la tecnica del *drag and drop*, richiedono di combinare i blocchi per dare istruzioni a un personaggio rispetto al percorso che deve seguire per raggiungere un determinato obiettivo (Kalelioğlu, 2015; Nardelli & Ventre, 2015). I personaggi sono basati su figure riconoscibili prese da cartoni animati o videogiochi, rendendo l'ambiente familiare. La programmazione visuale a



blocchi non richiede una conoscenza di linguaggi di programmazione, risultando quindi di facile comprensione anche da parte di insegnanti con pochissima formazione specifica in ambito informatico (Cetin, 2016; Nardelli & Ventre, 2015).

In uno studio del 2015, Kalelioğlu ha indagato l'effetto dell'utilizzo di Code.org sulle capacità di pensiero riflessivo, messe in atto durante il problem solving, coinvolgendo 32 alunni di classe quarta della scuola primaria. Oltre al confronto pre-test/post-test delle prestazioni su Code.org e delle capacità di pensiero riflessivo, dopo la fase sperimentale è stata fatta un'analisi qualitativa tramite focus group, per indagare l'interesse verso la programmazione, la difficoltà nello svolgere gli esercizi e la facilità di utilizzo della piattaforma. I risultati hanno mostrato che le attività di *coding* non hanno avuto influenza sulla capacità riflessiva, ma l'analisi qualitativa ha mostrato un atteggiamento positivo e di interesse verso la programmazione, sia da parte dei partecipanti maschi che femmine.

Un'altra piattaforma di programmazione a blocchi molto diffusa è Scratch che, a differenza di Code.org, in cui ci sono esercizi prestabiliti da risolvere, permette di creare e sviluppare liberamente giochi, storie e animazioni in modo intuitivo (Sáez-López et al., 2016). Ci sono evidenze che l'utilizzo di Scratch in modo strutturato porta non solo a un miglioramento significativo nei compiti di *coding*, supportando quindi lo sviluppo del pensiero computazionale, ma anche al trasferimento di queste conoscenze e abilità in ambienti di programmazione *text-based* (Grover et al., 2015). Inoltre, rispetto ai tradizionali metodi di insegnamento dell'informatica, utilizzando un approccio attivo e creativo come Scratch, si ottengono benefici in termini di motivazione, interesse, impegno e divertimento rispetto alla programmazione (Grover et al., 2015; Sáez-López et al., 2016). Questo vale non solo per gli studenti, ma anche per gli insegnanti che vengono formati sull'utilizzo di Scratch con i loro alunni (Cetin, 2016). In uno studio del

2016, Cetin ha messo a confronto l'effetto dell'apprendimento di concetti base di programmazione tramite Scratch con l'effetto ottenuto con metodo tradizionale tramite C, un linguaggio di programmazione con un campione di insegnanti. Il gruppo sperimentale, utilizzando Scratch, ha raggiunto una comprensione dei concetti base dell'informatica significativamente migliore rispetto al gruppo di controllo. Non è stata invece riscontrata una differenza tra gruppi nell'attitudine verso il *coding*.

In conclusione, la programmazione visuale a blocchi risulta sia uno strumento utile per sviluppare le abilità di programmazione, ma anche uno strumento che può essere integrato in modo interdisciplinare (Sáez-López et al., 2016) e questo ha importanti implicazioni per l'implementazione dello sviluppo del CT in ambito educativo.

### **1.3.2 Coding *Unplugged***

Le pratiche di *coding unplugged* sono un buon mezzo per introdurre, attraverso rappresentazioni concrete, concetti di informatica nel corso della prima infanzia, prima che i bambini inizino ad utilizzare i computer (Lee & Junoh, 2019). Le attività richiedono solitamente pochi materiali facilmente reperibili e sono quindi accessibili per chiunque (Barth-Cohen et al., 2019). Inoltre, il *coding unplugged* può essere efficacemente integrato con il regolare curriculum previsto per la scuola dell'infanzia, anche in associazione ad attività di robotica (Metin, 2022).

Secondo Bell et al. (2009) può essere uno strumento adatto per introdurre le basi della programmazione a tutte le età. In un recente studio (Dağ et al., 2023), un corso di *coding unplugged* si è rivelato essere efficace nel migliorare le abilità e i processi sottostanti al pensiero computazionale in alunni di classi terza e quarta della scuola primaria. Inoltre, le competenze degli alunni erano significativamente e positivamente correlate con le loro abilità di collaborazione e comunicazione.

Nella vita quotidiana, i bambini sperimentano e usano il CT in maniera inconsapevole ogni volta che svolgono una routine, seguendo una procedura *step by step*, ad esempio seguendo tutti i passaggi necessari per allacciarsi le scarpe (Lee & Junoh, 2019). È proprio a partire da azioni della vita quotidiana che gli insegnanti possono cogliere l'opportunità di far apprendere ai loro alunni i principi del CT, sviluppando la capacità di identificare ed etichettare i singoli passaggi di una procedura (Lee & Junoh, 2019).

Secondo Lee e Junoh (2019), un primo modo per introdurre il *coding unplugged* è identificare quei compiti giornalieri che hanno le caratteristiche di un algoritmo, ad esempio lavarsi le mani o mettersi il giubbotto, e chiedere al bambino quali step compie per eseguire quel determinato compito. Nell'identificare gli step l'insegnante deve incentivare l'alunno a trovare il modo più efficace per completare il compito, eliminando gli step superflui. Inoltre, aiutare il bambino ad individuare e correggere gli errori rende possibile l'introduzione del concetto di *debugging*. Una volta identificati gli step, si possono far disegnare al bambino, per avere una rappresentazione visiva di tipo sequenziale.

Un altro strumento utile è il *coding sheet* (Lee & Junoh, 2019; Metin, 2022), un foglio contenente una griglia in cui vengono definiti una casella di partenza, su cui solitamente è collocato un personaggio, e una casella di arrivo, in cui è presente un oggetto da raggiungere. L'obiettivo è identificare la sequenza di frecce, e quindi l'algoritmo, che permette al personaggio di raggiungere l'oggetto. La sequenza trovata viene riportata in uno spazio apposito sotto alla griglia.

Questa tipologia di esercizi è molto simile agli esercizi proposti su Code.org. Durante lo svolgimento degli esercizi l'insegnante può fornire feedback utilizzando

esplicitamente termini direzionali e termini quali *coding*, algoritmo, sequenza, loop per aiutare il bambino a familiarizzare con queste parole (Lee & Junoh, 2019).

Lo stesso tipo di esercizio può essere svolto, invece che su un foglio, su una grande griglia disposta sul pavimento, utilizzando oggetti tangibili. In questo modo il bambino ha la possibilità di fare *coding* in modalità cinestetica, camminando sulla griglia, come se fosse lui stesso il robot che deve seguire i comandi (Lee & Junoh, 2019). La stessa attività può essere proposta anche senza una griglia, scegliendo un punto di partenza e un punto di arrivo nell'aula (Barth-Cohen et al., 2019).

Le attività appena proposte possono essere definite come cross curriculari, in quanto contribuiscono all'apprendimento di concetti e abilità richieste in altre discipline, ad esempio le pratiche scientifiche e ingegneristiche, il pensiero logico e la capacità di scrivere e fornire istruzioni chiare e concise (Barth-Cohen et al., 2019).

## Capitolo 2. Differenze di genere nelle discipline STEM e nel *coding*

### 2.1 STEM e differenze di genere

L'acronimo STEM, in lingua inglese, sta per Scienza, Tecnologia, Ingegneria e Matematica, ovvero l'insieme delle discipline scientifico-tecnologiche. A livello globale, è documentata una sottorappresentazione delle donne nelle carriere e nell'istruzione di ordine superiore in ambito STEM, seppur in calo negli ultimi decenni (UNESCO, 2017; Wang & Degol, 2017).

I dati ISTAT (Istituto Nazionale di Statistica) relativi al 2021 riportano che, nella fascia d'età tra i 25 e i 34 anni, nonostante i livelli di istruzione tra le donne siano più elevati (23,1% delle donne, contro il 16,8% degli uomini), tra i laureati in discipline STEM esiste un importante divario di genere. Tra gli uomini la percentuale di laureati STEM è del 33,7% (un laureato su tre), mentre per le donne la percentuale scende al 17,6% (una laureata su sei).

Montuori et al. (2022) riportano che le due principali spiegazioni riguardo alle differenze di genere in questo campo riguardano, da una parte, le differenze biologiche e innate nelle capacità cognitive che stanno alla base della performance in ambito STEM; dall'altra, i fattori socioculturali. Ricerche recenti mostrano che le differenze di abilità legate al genere siano dovute alle interazioni bio-psico-sociali tra le predisposizioni biologiche e gli aspetti socioculturali (Miller & Halpern, 2014; Wierenga et al., 2019).

A supporto di questa ipotesi, Wierenga et al. (2019) hanno svolto uno studio longitudinale con lo scopo di indagare gli effetti del genere sulla struttura cerebrale corticale, in relazione allo sviluppo delle funzioni esecutive. I risultati hanno mostrato che, seppur per i maschi è presente una maggiore varianza nella struttura e nello sviluppo cerebrale rispetto alle ragazze, questo non è correlato con le prestazioni a livello cognitivo. Le differenze di genere osservate nelle FE non sono quindi legate allo sviluppo

cerebrale, suggerendo l'ipotesi che queste siano invece dovute a differenti esperienze e strategie, piuttosto che allo sviluppo neuro-biologico.

Wang e Degol, nella loro review del 2017, affermano che la sottorappresentazione femminile in ambito STEM è dovuta alla complessa interazione di sei fattori sottostanti: differenza di abilità, punti di forza, preferenze nella scelta della carriera, preferenze relative allo stile di vita, credenze sulle abilità specifiche nell'area STEM e stereotipi e pregiudizi di genere. Per quanto alcuni di questi fattori siano dovuti ad aspetti biologici, è plausibile ritenere che siano maggiormente i fattori socioculturali ad influenzare la scelta della carriera nelle donne, piuttosto che la sola biologia (Wang & Degol, 2017).

Tra i fattori socioculturali alla base del *gender gap* in ambito STEM ritroviamo credenze e aspettative dei genitori, così come il loro grado di istruzione e status socioeconomico, l'influenza del gruppo dei pari, le diseguaglianze nei sistemi educativi, le aspettative degli insegnanti, le risorse di apprendimento, le norme sociali e culturali rispetto alla parità di genere e gli stereotipi di genere trasmessi dai media (Montuori et al., 2022; UNESCO, 2017; Wang & Degol, 2017). Rilevanti sono anche i fattori psicologici quali autoefficacia, interesse e motivazione (UNESCO, 2017).

Tipicamente, le differenze di genere in ambito STEM sono maggiormente osservate a partire dagli ultimi anni di scuola primaria e oltre (Montuori et al., 2022). Allo stesso tempo, con l'avanzare dell'età, i bambini sviluppano maggiore consapevolezza di sé e della società, assimilando dal loro contesto credenze e stereotipi, inclusi quelli relativi al genere (Master, 2021). Anche se non in modo consapevole, i processi di formazione degli stereotipi emergono durante la scuola dell'infanzia, rafforzandosi durante la scuola primaria e secondaria di primo grado (Master, 2021). Allo stesso modo, in questa fascia d'età cominciano a svilupparsi in lieve misura gli atteggiamenti e le credenze di genere nei confronti della tecnologia (Sullivan & Bers, 2016).

Già a sei anni nei bambini emerge lo stereotipo che i maschi siano più bravi delle femmine nei compiti di robotica e programmazione, ma non nella matematica o la scienza. Questo è quanto è riportato da Master et al. (2017) in uno studio il cui scopo era quello di valutare l'efficacia di un intervento di sviluppo della motivazione STEM nelle bambine, tramite la programmazione robotica, ipotizzando un aumento di interesse e di autoefficacia a seguito di un'esperienza positiva. Le bambine sottoposte all'intervento hanno mostrato maggiori livelli di interesse verso la tecnologia e maggiore autoefficacia rispetto alle bambine del gruppo di controllo, alla pari dei partecipanti di genere maschile.

Analogamente, Sullivan e Bers (2016) hanno condotto uno studio pilota con 45 bambini della scuola dell'infanzia e fino alla classe seconda della scuola primaria. I bambini hanno svolto per otto settimane degli incontri di programmazione e di robotica. La ricerca ha indagato sia aspetti quantitativi, ovvero la performance nei compiti di programmazione e di robotica, che aspetti qualitativi, attraverso un protocollo sviluppato ad-hoc, con lo scopo di valutare le attitudini di genere e verso la tecnologia, nonché la presenza di differenze di genere nei risultati ottenuti a seguito degli incontri. I risultati non hanno mostrato differenze di genere tra le performance nei compiti di robotica e i compiti di programmazione a bassa difficoltà, ma i maschi hanno ottenuto performance migliori nei compiti di programmazione più difficili. È stata inoltre confermata l'ipotesi che già in questa fascia d'età (4-7 anni), i bambini iniziano ad assimilare l'opinione rispetto a quali strumenti tecnologici siano adatti ai maschi e quali alle femmine.

Gli stereotipi e le aspettative del gruppo sociale di appartenenza nell'ambito STEM hanno quindi profondi effetti sulla motivazione e sulla percezione di sé in ambito accademico (Master, 2021). Infatti, uno stereotipo negativo può portare i bambini a dubitare delle proprie capacità e disincentivare l'interesse verso le discipline STEM. Per questo, risulta molto importante introdurre i concetti base delle discipline STEM già in

tenera età, prima che gli stereotipi di genere vengano sviluppati e interiorizzati (Sullivan & Bers, 2016).

Nonostante queste evidenze, risulta ulteriormente necessario ampliare le conoscenze su come sviluppare interventi efficaci che possano supportare la parità di genere nei settori STEM (Master, 2021).

## **2.2 Coding e differenze di genere**

Ci sono pochi studi che analizzano le differenze di genere nel *coding* o nelle abilità di pensiero computazionale e i risultati che ne derivano sono contrastanti.

La review sistematica di Bati (2022) ha sintetizzato le evidenze sperimentali sul pensiero computazionale nel corso della prima infanzia, relativamente alle variabili *plugged* versus *unplugged*, età e genere. Per quanto l'età sia una variabile rilevante che influisce sullo sviluppo del pensiero computazionale, la review ha mostrato che in bambini dai 3 a 5 anni non sono presenti differenze di genere.

Nella stessa fascia d'età, uno studio di Sullivan e Bers (2013) ha rilevato la presenza di differenze di genere in bambini esposti per la prima volta al *coding* nello svolgimento di compiti di robotica tangibile e di programmazione. I maschi hanno ottenuto un punteggio medio più alto rispetto alle femmine nella maggior parte delle prove; tuttavia, solo in due aree la differenza si è rivelata essere statisticamente significativa: il corretto collegamento di materiali robotici e la programmazione condizionale. Al contrario, in nessuna delle aree le femmine hanno raggiunto una media statisticamente superiore rispetto ai maschi.

Questi risultati sono coerenti con uno studio successivo degli stessi autori, discusso precedentemente (Sullivan & Bers, 2016), in cui in bambini tra i 4 e i 7 anni, esposti per la prima volta al *coding*, sono state riscontrate differenze di genere a vantaggio dei maschi in compiti avanzati di programmazione.



Proprio a partire da questo studio (Sullivan & Bers, 2016), Montuori et al. (2022) hanno testato, con un campione di alunni di classe prima della scuola primaria, l'ipotesi che le differenze di genere osservate in giovane età nel *coding* siano dovute a differenze di genere nelle FE, in particolare nella pianificazione e nell'inibizione delle risposte, abilità strettamente legate al CT (Arfé et al., 2019, 2020). I risultati non hanno mostrato alcuna differenza di genere nelle prove di *coding* prima dell'inizio del corso introduttivo, mentre dopo il periodo di esposizione al *coding* è stata riscontrata una differenza di genere a vantaggio dei maschi. Inoltre, le differenze di genere nel *coding* non sono risultate essere mediate dalle FE dei bambini. Questo suggerisce che alla base di questo fenomeno ci siano altri fattori, come ad esempio il coinvolgimento in questo genere di attività, aspetti motivazionali o variabili socio cognitive.

In alunni di età superiore, le evidenze sono maggiormente discordanti. Alcuni studi con bambini tra i 9 e gli 11 anni circa non hanno riportato differenze di genere significative nei compiti di *coding* (Jiang & Wong, 2021; Kožuh et al., 2018; Price & Price-Mohr, 2023). Anche studenti di 8-17 anni sottoposti a corsi introduttivi di *coding* non hanno mostrato differenze statisticamente significative in termini di performance e miglioramento ottenuto nel *coding* (Papavlasopoulou et al., 2020). Tuttavia, sono emerse differenze di genere rispetto alle strategie di risoluzione utilizzate e all'attitudine verso le attività di *coding*.

Al contrario, Román-González et al. (2017), con un campione di studenti di 10-16 anni hanno trovato differenze di genere significative nei compiti legati al CT, a vantaggio dei ragazzi. Inoltre, hanno riscontrato che il *gender gap* cresce all'aumentare dell'età.

Anche Statter e Armoni (2017) hanno riscontrato differenze di genere in un campione di studenti al secondo anno di scuola secondaria di primo grado. Tuttavia, la

direzione dell'effetto di genere risulta opposta rispetto a quella trovata nello studio di Román-González et al. (2017). In questo caso, infatti, le ragazze hanno ottenuto un vantaggio significativamente maggiore rispetto ai maschi nei compiti di astrazione legati al *coding*. Inoltre, il contesto di insegnamento ha maggiormente e positivamente influito sull'apprendimento delle ragazze, come mostrato sia dalle migliori performance che dal maggiore interesse mostrato durante lo svolgimento del corso. Questo ha importanti implicazioni sulle modalità da applicare per contribuire a ridurre la sottorappresentazione femminile nel campo informatico. Un contesto di apprendimento stimolante potrebbe influenzare l'autoefficacia delle ragazze, incoraggiandole e motivandole a perseguire una carriera in questo ambito (Statter & Armoni, 2017).

In conclusione, date le evidenze contrastanti sulla presenza e sulla direzione delle differenze di genere nel *coding*, è possibile affermare che sono necessari maggiori studi per indagare questo fenomeno e contribuire a ridurre il *gender gap* in ambito STEM. Il presente studio si pone quindi come obiettivo quello di portare maggiori evidenze a riguardo, analizzando la presenza di un diverso effetto nel miglioramento delle abilità di *coding* tra maschi e femmine a seguito di un intervento di training.

## Capitolo 3. L'autoefficacia

### 3.1 Basi sull'autoefficacia

Il senso di autoefficacia, concetto introdotto dallo psicologo canadese Albert Bandura, “corrisponde alle convinzioni circa le proprie capacità di organizzare ed eseguire le sequenze di azioni necessarie per produrre determinati risultati” (Bandura, 2000, p. 23). Non si tratta della valutazione delle effettive abilità possedute, piuttosto di una percezione soggettiva, ovvero la convinzione personale 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).

La percezione di autoefficacia ha un'elevata rilevanza, influenzando in modo diretto la scelta di impegnarsi in un compito, lo sforzo messo in atto e la persistenza (Bandura, 2000; Mayall, 2008). Ne consegue che sia un valido predittore del comportamento e un elemento centrale del livello di motivazione. Inoltre, la sua definizione racchiude il concetto che le aspettative di successo rispetto al compito vengono espresse prima della sua esecuzione e sono influenzate da esperienze passate di risultati ottenuti (o meno) in compiti simili (Bandura, 2000; Moè, 2020).

Bandura (2000) individua tre dimensioni alla base dell'autoefficacia: la **generalità**, ovvero la misura in cui la percezione di autoefficacia viene generalizzata ed estesa a compiti simili; la **forza**, intesa come il grado di convinzione di ottenere il risultato atteso; il **livello**, ovvero il grado di percezione di controllo esperito.

La componente di percezione di controllo risulta molto importante. È definibile come la credenza di essere i principali responsabili del successo di un compito e avere quindi il controllo di determinare l'esito. Una maggiore percezione di controllo porta a maggiore autoefficacia e, di conseguenza, a maggiori aspettative di riuscita e impegno, aumentando le probabilità di successo (Bandura, 2000; Moè, 2020). Tuttavia, la

percezione di controllo non è l'unico elemento che entra in gioco quando si parla di autoefficacia: un ruolo importante è svolto anche dall'autoregolazione dei processi di pensiero, dalla motivazione e dagli stati affettivi e fisiologici (Bandura, 2000).

La percezione di efficacia si definisce come situazione-specifica, ovvero si esprime per un determinato compito, in un determinato momento ed è quindi un qualcosa di dinamico, che cambia in base all'esperienza, al contesto e ai fattori psicologici (Bandura, 2000; Moè, 2020). La stessa persona può, per compiti simili svolti in contesti diversi, mostrare differenti livelli di abilità. Lo stesso vale per persone diverse con capacità paragonabili. Questo dipende proprio dalle fluttuazioni nelle convinzioni di autoefficacia (Bandura, 2000).

### **3.1.1 Autoefficacia e concetti affini**

Il concetto di autoefficacia è affine ad altri costrutti psicologici, dai quali va distinto. Tra questi ritroviamo: la percezione di competenza, l'autostima, il *locus of control* e le aspettative di risultato.

Più in particolare, l'autoefficacia non va confusa con il concetto di percezione di competenza in quanto quest'ultima, al contrario dell'autoefficacia, è tendenzialmente stabile, si esprime a livello generale, anche durante o dopo l'esecuzione di un compito, ed è definibile come “un sentimento generale di fiducia nell'affrontare le cose” (Moè, 2020, p. 96).

Un'altra distinzione importante è tra l'autoefficacia e l'autostima, termini usati spesso in modo intercambiabile ma che in realtà si riferiscono a concetti diversi. Il concetto di autoefficacia si riferisce a un giudizio sulle proprie capacità, mentre l'autostima riguarda un giudizio sul valore personale: avere una stima di sé elevata non vuol dire per forza sentirsi capace di eseguire un compito e viceversa. Allo stesso modo,

un sentimento di autosvalutazione non sempre porta ad avere una scarsa autoefficacia (Bandura, 2000; Caprara, 2001).

Per *locus of control* (Rotter, 1966) si intende il grado di convinzione per cui gli eventi e i risultati ottenuti siano determinati dalle proprie azioni (locus of control interno) o da elementi esterni, al di fuori del proprio controllo (locus of control esterno). Anche in questo caso, avere un locus of control interno o esterno non coincide con un alto o basso senso di autoefficacia: ad esempio, si può ritenere che la vincita di una gara sportiva dipenda dalle proprie prestazioni, ma essere scoraggiati perché non si ritiene di essere in grado di affrontarla.

Infine, le aspettative di risultato sono legate alle conseguenze attese a seguito di una prestazione. In particolare, riguardano gli esiti fisici, sociali e gli effetti autovalutativi del comportamento messo in atto, aspetti ben diversi dal giudizio sulle capacità di esecuzione in una prestazione (Bandura, 2000; Caprara, 2001).

### 3.1.2 Sostenere l'autoefficacia

Il senso di autoefficacia può essere supportato e incrementato tramite differenti strategie, basate su informazioni provenienti da quattro fonti principali (Bandura, 1986, 2000):

1. **Esperienze dirette (o *mastery*):** consiste nel fare riferimento a situazioni in cui è stato portato a termine con successo un compito simile a quello che si sta per affrontare. Risulta una fonte molto potente di autoefficacia, se non la più potente. I successi passati, in cui un compito è stato padroneggiato, sono indicatori di capacità e possono produrre un elevato senso di autoefficacia, motivando ad impegnarsi anche in situazioni di difficoltà;
2. **Esperienze vicarie:** si basa sul concetto di modellamento o apprendimento osservativo. Consiste nell'osservare le esperienze di successo o insuccesso di

altre persone, che diventano modelli di riferimento. Tanto più vicini e simili si percepiscono i modelli, più i loro successi avranno un impatto. Non per forza un successo di un'altra persona porta a maggiore autoefficacia e un insuccesso porta ad un decremento: osservare una persona capace avere un'esperienza di insuccesso può, al contrario, incrementare il senso di autoefficacia se si ritiene di avere maggiori capacità o strategie più adatte a disposizione;

3. **Persuasione verbale:** si tratta dell'influenza dovuta alle affermazioni di incoraggiamento e fiducia di altre persone. Queste affermazioni incrementano maggiormente l'autoefficacia se provenienti da persone significative, ritenute vicine a sé o abili ed esperte nel compito in questione. In questo modo, la persona sarà più e più a lungo disposta ad impegnarsi nell'attività in oggetto. Le valutazioni positive non sono sufficienti di per sé nel determinare un cambiamento del senso di efficacia, lo sono solamente se realistiche. Al contrario, fornire incoraggiamenti irrealistici porta ad aumentare la probabilità di insuccessi, minacciando l'autoefficacia;
4. **Stati fisiologici e affettivi:** si basa sull'influenza delle informazioni somatiche provenienti da stati fisiologici e affettivi. Ad esempio, l'attivazione fisiologica dovuta allo stress è vista come un segno di debolezza e vulnerabilità e solitamente minaccia il senso di autoefficacia. Lo stesso vale per gli stati dell'umore, ad esempio quando si provano emozioni negative. Di conseguenza, un modo per sostenere l'autoefficacia è quello di migliorare le proprie condizioni fisiche e affettive e interpretare correttamente gli stati fisiologici.

### 3.1.3 Autoefficacia scolastica

Il senso di autoefficacia si esprime in diversi ambiti della propria vita. Tra questi rientra quello scolastico, di particolare rilevanza in giovane età. In un contesto di

apprendimento, maggiori convinzioni di autoefficacia portano ad un migliore funzionamento cognitivo, influenzando in modo diretto la carriera scolastica, nonché le aspirazioni professionali future (Bandura, 2000; Bandura et al., 2001; Bouffard-Bouchard, 1990).

L'obiettivo della scuola deve essere quello di promuovere credenze positive sul successo scolastico e sulle capacità di ogni studente, tramite un insegnamento di tipo interattivo, fornendo feedback correttivi e favorendo un modello di padronanza dell'apprendimento (Bandura, 2000). Particolare attenzione deve essere posta agli studenti che riscontrano maggiori difficoltà e che possono essere incoraggiati sia dagli insegnanti che da altri alunni, ad esempio lavorando in gruppo. Un altro modo per incoraggiare gli studenti è dividere le attività complesse in sotto obiettivi, rendendo il compito più facile da affrontare. Così facendo, lo studente si sentirà gratificato al raggiungimento di ogni sotto obiettivo e sarà più motivato a continuare il compito. Questo risulta importante perché le esperienze di fallimento a scuola possono avere conseguenze molto negative, con impatto anche in età adulta; al contrario, il successo didattico può contrastare un processo di sviluppo disadattivo (Caprara, 2001).

Ci sono evidenze rispetto al fatto che studenti con diversi livelli di autoefficacia ma stesse capacità differiscono sulle strategie utilizzate nella risoluzione di problemi, maggiormente adeguate per coloro che si percepiscono più capaci. Inoltre, studenti con alta autoefficacia affrontano le difficoltà con minore esitazione e, in generale, ottengono migliori risultati scolastici (Bandura, 2000; Bouffard-Bouchard, 1990; Caprara, 2001).

Bandura (2000) propone diverse strategie per migliorare l'autoefficacia degli studenti. Uno strumento utile è l'insegnamento individualizzato, che tiene conto delle diverse conoscenze e abilità di ogni alunno, limitando il confronto con gli altri in termini di valutazione, pratica che a volte può essere demotivante, con effetto di riduzione

dell'autoefficacia. In questo modo si ostacola la dipendenza degli alunni dalle opinioni di insegnanti e compagni.

Da incoraggiare è invece la cooperazione, soprattutto in classi eterogenee per cultura e profilo educativo. In questo risulta utile il tutoring fra pari, in cui uno studente più grande fa da modello ad uno studente più piccolo. Tramite questo strumento entrambe le parti ottengono benefici in termini didattici, di abilità sociali e di comunicazione (Bandura, 2000).

Uno strumento validato per la valutazione dell'autoefficacia scolastica è la Scala di Autoefficacia Scolastica Percepita che misura le convinzioni degli studenti rispetto alle proprie capacità di studiare determinate materie scolastiche, regolare la propria motivazione verso lo svolgimento delle attività didattiche, richiedere supporto per l'apprendimento e adottare modalità di studio adeguate (Pastorelli & Piccioni, 2001). La scala verrà descritta maggiormente nel prossimo capitolo.

### **3.2 Autoefficacia legata al *coding***

L'autoefficacia rispetto alla programmazione si può definire come la credenza personale di avere sufficienti competenze e abilità per eseguire adeguatamente un'attività di programmazione (Frymier et al., 1996).

È ritenuta da Kong et al. (2018) uno degli elementi fondamentali per lo sviluppo delle abilità di programmazione, insieme a significatività, impatto e autoefficacia legata alla creatività. Gli autori, a seguito della somministrazione di un questionario a 287 studenti di scuola primaria di età differenti, hanno riportato che studenti con maggiore interesse verso la programmazione mostrano anche maggiore autoefficacia, sia legata alla programmazione che alla creatività, e ne attribuiscono maggiore significatività e impatto. Quest'ultimo è definibile come il grado in cui la realizzazione di un compito è percepita influire e fare la differenza su uno schema più generale (Frymier et al., 1996). Un altro



risultato mostrato da Kong et al. (2018) è che l'autoefficacia legata alla creatività è maggiore in studenti più propensi alla collaborazione reciproca, in linea con la teoria di Bandura (2000) sulla cooperazione. Infine, alunni più grandi hanno mostrato bassa autoefficacia rispetto alla programmazione, ritenuta come meno significativa e i maschi hanno mostrato maggiore interesse nella programmazione rispetto alle femmine (Kong et al., 2018).

Rispetto alla cooperazione, Kim et al. (2021) hanno riscontrato che il tutoring tra pari favorisce la motivazione e l'autoefficacia verso le materie scientifiche, anche in giovani appartenenti a minoranze etniche, storicamente sottorappresentate in ambito STEM, sottolineando l'importanza di ricerche con prospettiva socioculturale.

Allo stesso tempo, le attività di sviluppo del pensiero computazionale, come il *coding*, possono favorire il senso di autoefficacia. Uno studio longitudinale di Fletcher et al. (2021) ha mostrato, in alunni di scuola primaria, un effetto positivo e immediato di un intervento volto a migliorare l'interesse e l'autoefficacia verso la programmazione, in particolare nell'ambito IT (Information Technology, ovvero Tecnologia dell'informazione), col fine ultimo di promuovere prospettive di carriera in questo campo. A distanza di qualche mese, gli effetti ottenuti sono scemati in modo significativo rispetto all'interesse, mentre per l'autoefficacia la diminuzione è stata minore e il punteggio risultava comunque maggiore rispetto alla misura baseline iniziale. Questo risultato fa ben sperare sugli effetti di incremento dell'autoefficacia attraverso attività simili a quelle descritte, ma sono necessari ulteriori studi longitudinali per confermare questa evidenza (Fletcher et al., 2021).

Ad influire sull'autoefficacia nel *coding* sono anche il tipo di attività proposte e il contesto di apprendimento (Güdel et al., 2019; Leonard et al., 2016; Statter & Armoni, 2017). In alunni della scuola secondaria di primo grado, Leonard et al. (2016) hanno

misurato la variazione del livello di autoefficacia dovuta ad attività di robotica e progettazione di videogiochi. I risultati hanno mostrato una diminuzione dell'autoefficacia rispetto all'utilizzo del computer, mentre per l'utilizzo di videogiochi è rimasta invariata. Tuttavia, sono state riscontrate delle differenze di autoefficacia in base all'ambiente di apprendimento: le attività di robotica combinate ad attività di sviluppo di videogiochi hanno portato a un incremento maggiore di autoefficacia rispetto ai videogiochi, a differenza delle sole attività di progettazione di videogiochi. Inoltre, nel contesto combinato i bambini hanno utilizzato differenti strategie di pensiero computazionale, con valutazioni più elevate. L'ambiente di apprendimento non ha invece influito sugli atteggiamenti in ambito STEM.

Mason & Rich (2020) hanno recentemente sviluppato e validato una scala che permette di valutare l'attitudine verso il *coding* in alunni della scuola primaria. La Scala ESCAS (Elementary Student Coding Attitudes Survey) misura i costrutti di interesse, utilità percepita, valore sociale, percezione dei programmatori e confidence, intesa come autoefficacia verso la programmazione. Questo strumento, utilizzato anche nel presente studio, può risultare utile agli insegnanti per misurare e comprendere quali variabili influenzano gli atteggiamenti dei bambini verso il *coding*, supportandone l'apprendimento.

### **3.2.1 Differenze di genere nell'autoefficacia legata al *coding***

In letteratura è stato riscontrato che le differenze di genere nell'autoefficacia percepita sono un'importante variabile che contribuisce al *gender gap* in ambito STEM (Bergey et al., 2015; Master, 2021; UNESCO, 2017). Tuttavia, gli studi che indagano la presenza di differenze di genere nell'autoefficacia STEM, e più in particolare, nel *coding* sono limitati e non sempre mostrano risultati concordanti.

Alcuni studi hanno analizzato la presenza di differenze di genere nell'autoefficacia legata al *coding* con campioni di studenti universitari. Uno studio di Busch (1995) ha indagato questo fenomeno in relazione ad autoefficacia e attitudine verso i computer. Rispetto alle femmine, i maschi hanno riportato di aver avuto maggiore esperienza pregressa nella programmazione e nei videogiochi su pc, oltre che maggiori incoraggiamenti da genitori e amici verso l'ambito informatico. Inoltre, i maschi hanno mostrato maggiori livelli di autoefficacia per compiti complessi di programmazione, mentre per compiti più facili non è stata riscontrata alcuna differenza di genere.

In uno studio più recente, svolto ugualmente su studenti universitari di Computer Science, Lishinski et al. (2016) hanno esaminato l'interazione tra l'autoefficacia, le strategie metacognitive e l'atteggiamento verso gli obiettivi di tipo intrinseco ed estrinseco, nonché il loro impatto sulle performance ottenute in un corso di informatica. I risultati hanno mostrato che le variabili considerate influiscono sulla performance che a sua volta influisce sull'autoefficacia. Tuttavia, sono state riscontrate differenze di genere nella relazione tra autoefficacia e performance. In particolare, le femmine hanno modificato le loro convinzioni di autoefficacia in base ai feedback sulla performance in una fase iniziale del corso, precedentemente rispetto ai maschi.

Nello studio di Bergey et al. (2015), invece, non è stata riscontrata la presenza di differenze di genere nel modo in cui l'autoefficacia influisce sulla performance, mentre per l'autoefficacia verso i videogiochi sono stati mostrati livelli maggiore nei maschi. Gli autori hanno esaminato possibili differenze di genere in studenti di scuola secondaria di primo grado relativamente a un particolare tipo di autoefficacia in ambito scientifico, ovvero quella legata alla ricerca scientifica. In particolare, lo scopo dello studio era verificare se e in che modo la performance in un compito in ambito scientifico eseguito in un ambiente virtuale immersivo fosse associata alle credenze di autoefficacia legata

alla ricerca scientifica, sia precedentemente che successivamente allo svolgimento del compito. Inoltre, lo studio ha indagato il ruolo dell'autoefficacia rispetto ai videogiochi e del genere sulla performance. I risultati hanno indicato l'autoefficacia legata alla ricerca scientifica come predittore della performance, che a sua volta ha influito sui cambiamenti di autoefficacia, mentre lo stesso non vale per l'autoefficacia rispetto ai videogiochi. Non sono state riscontrate differenze di genere significative nella performance e nell'autoefficacia legata alla ricerca scientifica, mentre per l'autoefficacia legata ai videogiochi i maschi hanno mostrato livelli maggiori.

La relazione tra autoefficacia e performance è coerente tra i due studi appena mostrati (Bergey et al., 2015; Lishinski et al., 2016), ma lo stesso non si può dire per la presenza di differenze di genere in questa relazione.

Un altro studio svolto su studenti di scuola secondaria di primo grado è quello di Güdel et al. (2019). Gli autori hanno mostrato che i livelli di interesse, autoefficacia e aspirazioni di carriera legati all'ambito della tecnologia cambiano a seconda del tipo di attività proposte. Inoltre, le aspirazioni di carriera si sono mostrate essere fortemente collegate all'autoefficacia, più che all'interesse verso la tecnologia. Relativamente alle differenze di genere, i maschi hanno mostrato in modo significativo un maggiore interesse e una maggiore autoefficacia rispetto alla tecnologia per gli item riferiti nello specifico ad attività più tecniche e inerenti all'ambito tecnologico.

Uno studio di Tellhed et al. (2022) con studenti svedesi di 12 anni ha indagato la presenza di differenze di genere in autoefficacia e 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, in contrasto con quanto riportato da Kong et al. (2018), che hanno trovato una differenza di genere solo rispetto all'interesse, e da Mayall (2008) che non ha rilevato differenze di genere nell'autoefficacia verso la tecnologia. Nello studio di Tellhed et al. (2022), la differenza di genere riscontrata è risultata essere completamente mediata dall'esperienza pregressa di utilizzo del computer, dalla persuasione verbale e dall'esperienza vicaria, ovvero le fonti di autoefficacia indicate da Bandura (1986, 2000). Dati i risultati contrastanti con altre ricerche, Tellhed et al. (2022) riportano come possibile spiegazione il diverso contesto culturale in cui queste sono state svolte, fattore che è dimostrato essere rilevante nella formazione di stereotipi, che a loro volta potrebbero influire sul senso di autoefficacia rispetto al *coding* (Bergey et al., 2015).

In conclusione, dato il numero limitato di evidenze e risultati talvolta contrastanti, è possibile affermare che sono necessari maggiori studi che indaghino la presenza di differenze di genere nell'autoefficacia legata al *coding* in culture e fasce d'età diverse, ad esempio nella scuola primaria, così da contribuire all'obiettivo di comprendere al meglio questo fenomeno.

## Capitolo 4. Metodo

### 4.1 Obiettivo di ricerca

Lo studio proposto nasce a partire dal 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.

Studi precedenti hanno dimostrato l'efficacia di un training di *coding*, sia sulle abilità di *coding* stesse che sul potenziamento delle funzioni esecutive in alunni di classi prime della scuola primaria (Arfé et al., 2019, 2020). Nel recente studio di Montuori et al. (2022), svolto con alunni di età media di circa 6 anni, è stato rilevato che l'esposizione a un training di *coding* porta ad un miglioramento delle abilità di *coding* in misura diversa per maschi e femmine. Prima del training, infatti, non sono state riscontrate differenze di genere nel punteggio di accuratezza al *coding*, mentre successivamente al training i maschi hanno raggiunto un livello di accuratezza significativamente maggiore rispetto alle femmine.

A partire da queste evidenze, il presente studio si è posto come obiettivo quello di indagare la presenza di una differenza di genere nelle abilità di *coding* in un campione di alunni di età maggiore, nello specifico di classi quarte della scuola primaria, esaminando il ruolo dell'autoefficacia su questo effetto.

Le ipotesi di ricerca erano:

1. Osservare un incremento significativo dei punteggi di accuratezza nelle prove di *coding* a seguito dell'intervento di training;
2. Che i maschi avrebbero ottenuto un maggiore miglioramento in accuratezza nelle prove di *coding* rispetto alle femmine a seguito dell'intervento di training;
3. Si sarebbe osservata una differenza di genere nei punteggi di autoefficacia STEM e di autoefficacia legata al *coding*;

4. Maggiori punteggi di autoefficacia STEM e di autoefficacia legata al *coding* sarebbero stati correlati a un maggiore miglioramento dei punteggi di accuratezza nelle prove di *coding* a seguito dell'intervento di training.

#### 4.1.1 Disegno sperimentale

Il disegno sperimentale del presente studio è di tipo misto. Presenta infatti sia fattori *between* che fattori *within*. In particolare:

- La variabile gruppo (sperimentale o di controllo) (ipotesi 1) e la variabile genere (ipotesi 2, 3 e 4) rappresentano il fattore *between*. I risultati del gruppo sperimentale sono stati confrontati con i risultati del gruppo di controllo. Allo stesso modo, i risultati ottenuti dai maschi sono stati confrontati con i risultati ottenuti dalle femmine.
- La variabile tempo (T1, T2, T3) rappresenta il fattore *within*. Sono stati confrontati i risultati ottenuti dai soggetti del gruppo sperimentale a T1 con i risultati ottenuti a T2 e a T3. Allo stesso modo, sono stati confrontati i risultati ottenuti dal gruppo di controllo nei tre diversi tempi, così come quelli ottenuti dai maschi e dalle femmine.

Il gruppo di controllo è di tipo *waiting list*. Questo vuol dire che, nel momento in cui il gruppo sperimentale riceve l'intervento, ovvero il training di *coding*, il gruppo di controllo non riceve alcun intervento, in questo caso proseguendo con le normali attività didattiche, e viene sottoposto all'intervento solo in una fase successiva.

Il vantaggio di questa tipologia di gruppo di controllo è quello di poter osservare l'effetto dell'intervento nel gruppo sperimentale, confrontando i risultati ottenuti con quelli del gruppo di controllo. È così possibile determinare l'efficacia dell'intervento confrontando gli effetti del training con quelli dovuti ai fisiologici processi di sviluppo.

Per far sì che anche i soggetti appartenenti al gruppo di controllo riscontrino i benefici dovuti all'intervento, viene proposto il training in un secondo momento.

Le attività sono state svolte in più fasi a partire da dicembre 2022 fino a maggio 2023, in particolare:

- **T1 (pre-test):** in questa fase sono state valutate le funzioni esecutive e le abilità preliminari di *coding* sia del gruppo sperimentale che del gruppo di controllo. Sono stati inoltre somministrati la Scala di Autoefficacia Scolastica Percepita (Pastorelli & Piccioni, 2001) e il Questionario sulle teorie dell'intelligenza (Cornoldi et al., 2005);
- **Training per il gruppo sperimentale:** subito dopo la fase di pre-test, il gruppo sperimentale è stato sottoposto al trattamento che consiste in otto incontri di training sul *coding*, con lo scopo di potenziare le abilità di pensiero computazionale. Nel frattempo, il gruppo di controllo ha proseguito con le normali attività didattiche;
- **T2 (post-test):** dopo la fase di intervento per il gruppo sperimentale, sono state valutate nuovamente le funzioni esecutive e le abilità di *coding*, così come nella fase T1, sia del gruppo sperimentale che del gruppo di controllo. Inoltre, nel gruppo sperimentale è stata somministrata la Scala ESCAS (Elementary Student Coding Attitudes Survey) (Mason & Rich, 2020). Questo ha permesso il confronto tra il gruppo che riceve il trattamento e il gruppo che non riceve alcun trattamento, così da determinare quali effetti sono dovuti al training e quali ai fisiologici processi di sviluppo;
- **Training per il gruppo di controllo:** a seguito della fase di post test, il gruppo di controllo è stato sottoposto al training nelle stesse modalità utilizzate per il gruppo sperimentale. Questo fa in modo che anche i soggetti del gruppo di controllo



possano beneficiare degli effetti delle attività di *coding*. Nel frattempo, il gruppo sperimentale ha proseguito con le normali attività didattiche;

- **T3 (post-test):** al termine della fase di training per il gruppo di controllo, sono state valutate le abilità di *coding* per entrambi i gruppi. Inoltre, nel gruppo di controllo è stata somministrata la Scala ESCAS (Mason & Rich, 2020). Questo ha permesso di verificare l'effetto del trattamento nel gruppo di controllo e il mantenimento di questo effetto a distanza di tempo nel gruppo sperimentale.

## 4.2 Partecipanti

Lo studio è stato condotto su due classi quarte di una scuola primaria del territorio di Padova, per un totale di 44 partecipanti. Nel campione finale sono stati inclusi 43 partecipanti, di cui 22 appartenenti al gruppo sperimentale (51,2%) e 21 al gruppo di controllo (48,8%). Il campione include 17 femmine (39,5%) e 26 maschi (61,5%).

Insieme al consenso informato di adesione allo studio è stato somministrato ai genitori dei partecipanti un breve questionario sociodemografico con lo scopo di stimare il loro Status Socioeconomico (SES) e il livello di familiarità con la tecnologia (Fam Tech).

Per il calcolo del SES sono stati chiesti a entrambi i genitori il titolo di studio più alto conseguito e la professione. Le due componenti sono poi state trasformate in punteggi ed è stato sommato il punteggio più alto tra quelli relativi al titolo di studio con quello più alto relativo alla professione.

Il livello di familiarità con la tecnologia è stato calcolato sommando i punteggi ottenuti negli item in cui venivano richieste informazioni riguardo alla frequenza e alla modalità di utilizzo di dispositivi quali pc, tablet e smartphone.

I dati demografici sono riportati nella **Tabella 1**.

**Tabella 1.**

*Medie, deviazioni standard e confronto tra gruppi per le variabili età, SES, familiarità con la tecnologia, ToL, Stroop (errori) e Coding a TI divisi per gruppo e per genere.*

Variabile	Gruppo di controllo				Gruppo sperimentale			
	Femmine (n = 9)	Maschi (n = 12)	W	p	Femmine (n = 8)	Maschi (n = 14)	W	p
	M (SD)	M (SD)			M (SD)	M (SD)		
Età (mesi)	113.56 (4.28)	112.92 (4.32)	60	0.694	111.38 (2.88)	112.29 (3.47)	49.5	0.68
SES	5.56 (1.51)	5.82 (1.66)	44	0.698	5.88 (1.36)	5.7 (1.65)	53	0.97
Fam Tech	1.89 (0.6)	1.92 (0.79)	53.5	1	1.63 (0.92)	1.64 (0.84)	56.5	1
ToL	6.11 (3.02)	8.58 (2.39)	26.5	0.053	8.25 (2.6)	8.21 (2.94)	49.5	0.678
Stroop	4 (2.24)	3.17 (2.17)	65	0.448	2.75 (4.1)	4.29 (3.54)	38.5	0.236
Coding	0.33 (1)	1.17 (1.8)	37.5	0.154	0.63 (1.06)	0.21 (0.58)	69	0.244

*Nota.* I dati sul SES si riferiscono a 41 soggetti, poiché per un maschio del gruppo di controllo e un maschio del gruppo sperimentale non è stato possibile il calcolo del punteggio.

## 4.3 Strumenti

### 4.3.1 Funzioni Esecutive

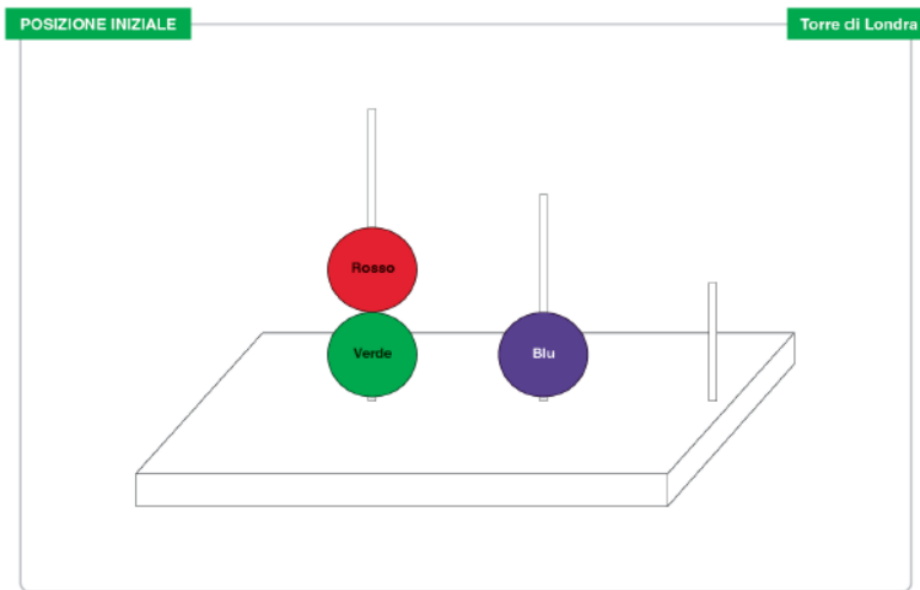
Le Funzioni Esecutive dei partecipanti sono state valutate utilizzando due test standardizzati, la Torre di Londra (ToL) (Fancello et al., 2006) e lo Stroop Numerico (Marzocchi et al., 2010) che misurano rispettivamente le abilità di pianificazione e le abilità di inibizione della risposta.

#### **Abilità di pianificazione**

La ToL è un test standardizzato che permette di valutare le abilità di pianificazione e di problem solving. Il suo svolgimento richiede competenze specifiche nell'ambito delle funzioni esecutive, quali la rappresentazione mentale del compito da eseguire, la pianificazione della sequenza di operazioni necessarie e l'inibizione e/o il rimandare le risposte impulsive. La versione usata in questo studio è quella standardizzata per bambini dai 4 ai 13 anni (Fancello et al., 2006).

Il test richiede di riprodurre una configurazione di tre palline colorate (verde, rossa, blu) poste su tre bastoncini di altezze diverse, rispettando le seguenti regole:

- Partire sempre dalla posizione iniziale (**Figura 2**);
- Muovere una sola pallina alla volta usando una sola mano;
- Non poggiare la pallina sul tavolo o tenerla in mano dopo averla presa per spostarne un'altra. Ogni volta che viene presa una pallina va infilata in un bastoncino;
- Sul bastoncino piccolo ci sta massimo una pallina, su quello medio massimo due palline, su quello grande massimo tre palline;
- Per ogni problema viene indicato al bambino un numero massimo di mosse da rispettare.



**Figura 2.** *ToL, posizione iniziale (Fancello et al., 2006)*

Il test è composto da 12 item che richiedono un numero di mosse sempre maggiori per essere completati. All'inizio è presente un item di esempio per verificare che il bambino abbia compreso tutte le regole.

Nel caso in cui il bambino si accorga di aver violato qualche regola o di aver sbagliato l'ordine in cui spostare le palline, può richiedere allo sperimentatore di ricominciare la prova, ritornando alla posizione iniziale.

Per ogni prova viene misurato il tempo necessario per risolvere il compito, dato dalla somma di:

1. Tempo di pianificazione: dal momento in cui viene mostrata la configurazione da copiare fino a quando la prima pallina viene completamente estratta dal bastoncino;
2. Tempo di esecuzione: dal momento dell'estrazione della prima pallina fino al completamento del compito.

Il punteggio assegnato ad ogni item è di 1 se questo viene risolto correttamente entro un minuto di tempo, senza violare alcuna regola. In caso contrario il punteggio è 0.

### **Abilità di inibizione della risposta**

Per valutare le abilità di inibizione della risposta è stato somministrato lo Stroop Numerico, un test standardizzato per bambini dai 6 agli 11 anni, presente all'interno della Batteria Italiana per l'ADHD (Marzocchi et al., 2010).

Il test è costituito da una prova baseline e dalla prova di Stroop, somministrate in successione. Il compito richiesto è dire il più velocemente possibile il numero di elementi contenuti in ogni singola cella mostrata. Le due prove sono così strutturate:

1. Prova baseline: viene mostrato un foglio con 12 caselle contenenti degli asterischi. Il numero di elementi presenti in ogni cella va da 1 a un massimo di 5;
2. Prova di Stroop: vengono mostrate complessivamente 75 celle, ciascuna contenente delle cifre, da 1 a 5, uguali tra loro. Anche in questo caso, il numero di elementi presenti in ogni cella va da 1 a 5.

Nella prova di Stroop, quindi, il bambino deve riportare la quantità degli elementi, senza farsi distrarre dalla loro identità. Riportare l'identità delle cifre sarebbe infatti la risposta più automatica, ma in questo caso deve essere inibita per fornire invece una risposta secondaria non automatica, ovvero la quantità di cifre.

In entrambe le prove vengono presi in considerazione il tempo necessario per completare la prova e il numero di errori commessi. Nella prova di Stroop vengono considerati sia gli errori di interferenza, in cui viene riportata l'identità dello stimolo e non la numerosità, sia gli errori di conteggio, in cui la risposta non coincide né con l'identità né con la numerosità dello stimolo. Vengono inoltre considerati il numero di

autocorrezioni e il numero di omissioni. L'accuratezza è rappresentata in termini di numero di errori e autocorrezioni.

#### **4.3.2 Misure di autoefficacia**

Le misure di autoefficacia sono state raccolte tramite la Scala di Autoefficacia Scolastica Percepita (Pastorelli & Piccioni, 2001) e l'Elementary Student Coding Attitudes Survey (Mason & Rich, 2020).

##### **Scala di Autoefficacia Scolastica Percepita (ASCP)**

La Scala ASCP (Pastorelli & Piccioni, 2001) è uno strumento utile ad indagare le convinzioni degli studenti rispetto alle loro capacità in ambito scolastico. La versione originale è formata da 19 item a scelta multipla. Il punteggio per ogni item va da 1 (per nulla capace) a 5 (del tutto capace).

Per il seguente studio sono stati considerati gli item da 1 a 7 in cui è richiesto di indicare quanto il partecipante si sente bravo nell'imparare alcune materie scolastiche, in particolare:

- Quanto ti senti bravo/a?
  1. Nell'imparare la matematica;
  2. Nell'imparare la geografia;
  3. Nell'imparare le scienze;
  4. Nell'imparare l'italiano;
  5. Nell'imparare la grammatica;
  6. Nell'imparare la storia;
  7. Nell'imparare le lingue straniere.

## Elementary Student Coding Attitudes Survey (ESCAS)

La Scala ESCAS, sviluppata da Mason e Rich (2020), misura le attitudini verso il *coding* in alunni della scuola primaria. È formata da 23 item su scala Likert a 6 punti che indagano 5 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. La scala è stata tradotta per poter essere somministrata in questo studio. Nella traduzione, la parola programmazione è stata affiancata al termine *coding*, data la familiarità acquisita dai partecipanti nel corso del training con il termine inglese. Sono stati somministrati 15 su 23 item, in particolare gli item delle sottoscale Confidence, Interesse e Utilità Percepita:

- **Confidence:**

1. Posso imparare a programmare/fare coding;
2. Sono bravo/a a programmare/fare coding;
3. Sono bravo/a 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 bravo/a a programmare/fare coding.

- **Interesse:**

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

- **Utilità percepita:**

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/coding per il mio futuro lavoro.

In aggiunta, sono state somministrate delle domande aperte con riquadri di risposta breve, con lo scopo di verificare un'adeguata comprensione sulla definizione di *coding*, termine poco familiare prima dell'intervento:

1. Secondo te, cos'è il coding/programmazione?
2. Che tipo di persone sono brave a fare coding/programmazione?
3. Secondo te, cos'è l'informatica?
4. Secondo te, che cosa fa a lavoro un informatico/a?

#### **4.3.3 Questionario sulle teorie dell'intelligenza**

Il Questionario sulle teorie dell'intelligenza (QC1I), presente all'interno della batteria AMOS 8-15 (Cornoldi et al., 2005), è una scala formata da 4 item, misurabili su scala Likert a 4 punti. Sono presenti quattro affermazioni riguardo all'intelligenza e il bambino deve indicare il grado di accordo (1 = d'accordo; 4 = contrario, con punteggi invertiti per gli item 3 e 4). Gli item sono i seguenti:

1. La tua intelligenza è qualcosa di te che non puoi cambiare;
2. Puoi imparare cose nuove, ma non puoi cambiare la tua intelligenza;
3. Indipendentemente da quanta intelligenza tu abbia, puoi sempre cambiarne almeno un po';
4. Quando impari cose nuove migliori anche la tua intelligenza.



#### 4.3.4 Abilità di *coding*

Le abilità di *coding* sono state valutate tramite una serie di esercizi svolti su Cothi.it, una piattaforma ispirata a Code.org. Subito dopo lo svolgimento del test è stato somministrato un questionario *ad hoc* di gradimento.

Il test sulle abilità di *coding* consiste in quattro prove di difficoltà crescente, ognuna preceduta da una prova di esempio. In ogni prova è richiesto di creare un codice individuando la sequenza corretta di blocchi che consente al personaggio (ad esempio l'uccellino rosso) di raggiungere uno o più obiettivi (ad esempio il maialino verde) seguendo il percorso più breve possibile.

Come si può vedere nella **Figura 3**, 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 feedback 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.

Per ogni esercizio vengono misurati:

- Tempo di pianificazione: dal momento in cui viene avviato l'esercizio a quando viene cliccato il primo blocco da spostare;

- Tempo di esecuzione: dal momento in cui viene spostato il primo blocco finché non viene cliccato il tasto “Esegui”;
- Accuratezza: il punteggio assegnato ad ogni esercizio, incluse le prove di esempio, è di 2 se questo viene risolto correttamente al primo tentativo; punteggio di 1 se viene risolto al secondo tentativo; punteggio di 0 se viene risolto al terzo tentativo o non viene risolto.

L’esempio della prima prova richiede di guidare l’uccellino rosso verso il maialino verde in un percorso lineare (**Figura 3**). I blocchi a disposizione sono:

- Vai avanti: il personaggio si sposta in avanti di una casella, nella direzione in cui sta guardando;
- Gira a sinistra/destra: il personaggio ruota su se stesso verso la propria sinistra/destra, senza spostarsi di casella.

Questo primo esercizio di familiarizzazione richiede l’uso esclusivo del blocco “vai avanti” per la corretta risoluzione.



**Figura 3.** Esempio prova 1 del test di coding su Cothi.it

Nella prima prova (**Figura 4**) ci sono due possibili percorsi da seguire per raggiungere l’obiettivo. In entrambi i casi è richiesto di utilizzare, oltre al blocco “vai

avanti”, anche i blocchi “gira a sinistra” e “gira a destra”. Questo esercizio permette quindi una familiarizzazione con i percorsi che presentano delle svolte.



**Figura 4.** Prova 1 del test di coding su Cothi.it

Nell’esempio della seconda prova (**Figura 5**) viene introdotto il blocco “ripeti”. Questo blocco richiede l’inserimento di uno o più blocchi al suo interno, ovvero l’azione da eseguire, indicando quante volte questa azione deve essere ripetuta. In questo caso la risoluzione richiede l’utilizzo del blocco “vai avanti” per cinque volte. L’esercizio è programmato perché sia possibile prendere un solo blocco “vai avanti”, quindi l’unico modo per risolverlo è utilizzando il blocco “ripeti” che presenta già all’interno il numero 5 come suggerimento. Il percorso, quindi, non presenta svolte ed è possibile superare la prova utilizzando esclusivamente i blocchi “vai avanti” e “ripeti”.



**Figura 5.** Esempio prova 2 del test di coding su Cothi.it

La prova 2 (**Figura 6**) richiede di far arrivare lo zombie al girasole, utilizzando il blocco “ripeti” in un percorso in cui sono presenti delle svolte. Nel percorso sono inseriti anche degli ostacoli, le piante carnivore. Anche in questo caso ci sono due possibili percorsi, ma uno risulta più breve dell’altro ed è quindi preferibile. Al contrario della prova di esempio, la risoluzione di questo esercizio richiede l’utilizzo di più blocchi all’interno del blocco “ripeti”, ovvero più blocchi “vai avanti” e il blocco “gira a destra”.



**Figura 6.** Prova 2 del test di coding su Cothi.it

La terza prova di esempio (**Figura 7**) richiede un’attività di *debugging*, cioè l’individuazione e correzione di errori presenti nel codice già fornito dal programma. L’ape deve raggiungere più obiettivi, ovvero passare sopra a tutti i fiori. Ogni volta che l’ape si ferma su uno dei fiori deve raccogliere tutto il nettare. Per rendere possibile ciò è stato introdotto il blocco “prendi il nettare”. Il numero di blocchi da utilizzare è pari al numero posto vicino al fiore che indica la quantità di nettare presente. Come si vede nella figura, tra i blocchi è presente anche “fai il miele”, non rilevante ai fini del test, in quanto è necessario solamente in alcune tipologie di esercizi svolti durante la fase di training, non presenti in questa prova. La risoluzione dell’esercizio prevede l’aggiunta del blocco “ripeti” e l’eliminazione di tre dei blocchi già presenti.



**Figura 7.** Esempio prova 3 del test di coding su Cothi.it

Così come nell'esercizio di esempio, nella prova 3 è richiesta l'abilità di *debugging* (**Figura 8**). Questa volta il percorso è più lungo e richiede due svolte. La risoluzione dell'esercizio consiste nell'aggiungere un blocco "prendi il nettare" e sostituire il secondo blocco "gira a sinistra" con il blocco "gira a destra".



**Figura 8.** Prova 3 del test di coding su Cothi.it

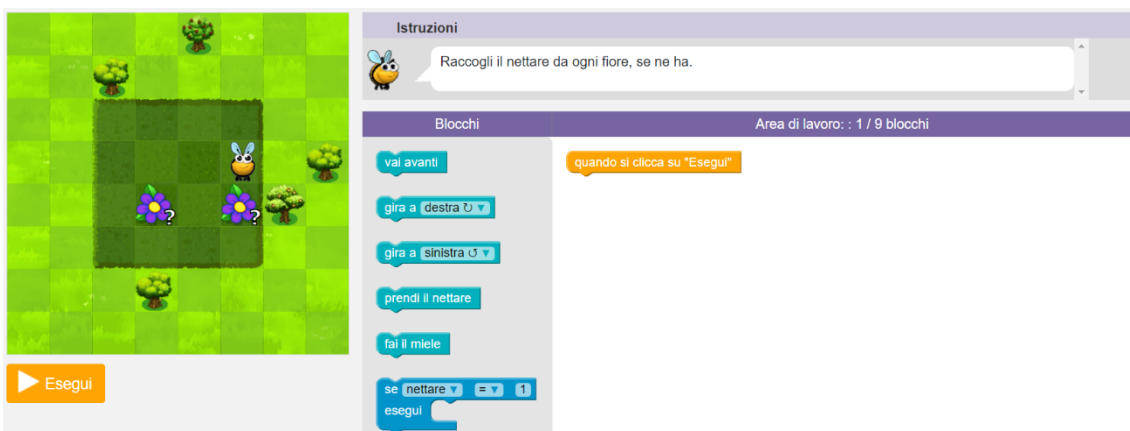
Nella quarta prova di esempio (**Figura 9**) viene introdotto il blocco "se". Questo blocco è necessario quando sono presenti fiori viola, la cui quantità di nettare può essere zero o uno e viene assegnata casualmente ogni volta che viene premuto il tasto "Esegui". Inserendo all'interno del blocco "se" il blocco "prendi il nettare", l'azione verrà eseguita

solamente se la quantità di nettare sarà pari a 1, in caso contrario non verrà svolta alcuna azione. L'esercizio è programmato per far sì che l'unico modo per raccogliere il nettare dai fiori viola sia utilizzando il blocco "se". Infatti, se viene usato solamente il blocco "prendi il nettare", il fiore viola non avrà nettare e l'esercizio non viene superato perché l'azione non può essere eseguita. Al contrario, se non viene usato il blocco "prendi il nettare", il fiore viola avrà del nettare che non potrà quindi essere raccolto.



**Figura 9.** Esempio prova 4 del test di coding su Cothi.it

Nella quarta e ultima prova (**Figura 10**) sono presenti due fiori viola. La risoluzione richiede quindi l'utilizzo di due blocchi "se" in un percorso che presenta una svolta.



**Figura 10.** Prova 4 del test di coding su Cothi.it

## Questionario di gradimento

A seguito della prova di *coding* viene somministrato un questionario *ad hoc* di gradimento che indaga il livello di piacevolezza del gioco (**Figura 11**).



**Figura 11.** Esempio di item del questionario di gradimento

Il questionario è composto da cinque item:

- 1) Quanto ti sei divertito/a facendo questa attività?
- 2) Quanto è stato facile fare queste attività?
- 3) Quanto saresti contento/a di fare altre attività come queste?
- 4) Quanto è stato difficile fare queste attività?
- 5) Ti senti bravo/a a fare queste attività?

Per ogni domanda viene richiesto di indicare un punteggio da 1 a 5:

- 1) Per niente
- 2) Poco
- 3) Abbastanza
- 4) Molto
- 5) Moltissimo

## 4.4 Procedura

Le attività sono iniziate a dicembre 2022 con la fase di pre-test (T1), sia per il gruppo sperimentale che per il gruppo di controllo. I test sono stati svolti in un'aula messa a disposizione dalla scuola, sufficientemente spaziosa e silenziosa e sempre in modalità individuale. In un primo momento sono state valutate le abilità di *coding* preliminari su pc portatile tramite la piattaforma Cothi.it, con cui è stato somministrato subito dopo il questionario di gradimento. A distanza di qualche settimana sono stati somministrati i test cognitivi ToL e Stroop Numerico per la valutazione delle Funzioni Esecutive, insieme ai questionari ASCP e QCII, somministrati solo in questa fase come predittori delle abilità di *coding*.

Nel mese successivo alla fase di pre-test si è svolta la fase di training per il solo gruppo sperimentale. Questa fase ha previsto otto incontri che si sono svolti collettivamente nell'aula informatica del plesso scolastico, dotata di computer fissi e proiettore. Ogni alunno aveva a disposizione un computer per svolgere gli esercizi in autonomia. All'inizio di ogni incontro venivano brevemente descritti gli obiettivi da raggiungere e spiegate le modalità di utilizzo di eventuali nuovi blocchi. I bambini sono stati incoraggiati a risolvere gli esercizi individualmente, leggendo sempre le istruzioni e riflettendo bene su come utilizzare il minor numero possibile di blocchi. In questa fase non è stato definito un numero massimo di tentativi. In caso di bisogno, 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 il bambino a raggiungere la soluzione in autonomia. Dopo il superamento delle prove, soprattutto quelle più complesse o che richiedevano l'utilizzo di un nuovo blocco, è stato lasciato spazio a una discussione collettiva, in cui due o tre bambini o bambine mostravano alla classe il modo in cui avevano risolto l'esercizio. L'obiettivo era sempre



quello di mettere in evidenza che per uno stesso problema si possono trovare strategie di risoluzioni differenti, ma anche portare i bambini a riflettere e trovare insieme la soluzione più efficiente, nel caso in cui nessuno l’avesse trovata autonomamente. I contenuti e gli obiettivi specifici per ogni incontro di training sono riportati nella **Tabella 2**.

**Tabella 2.** *Contenuti e obiettivi degli incontri di training.*

<https://studio.code.org/s/course2>; <https://studio.code.org/s/course3>.

Sessione	Corso	Esercizi	Contenuti	Obiettivi
Incontro 1	Corso 2 Lezione 3	2, 3, 6, 9	Labirinto: Sequenze	Percorsi lineari e con svolte
Incontro 2	Corso 2 Lezione 6	3, 6, 9, 10	Labirinto: Cicli	Introduzione blocco “ripeti”
Incontro 3	Corso 2 Lezione 8	1, 3, 6, 8	Ape: Cicli	Blocco “ripeti” con i fiori
Incontro 4	Corso 2 Lezione 10	2, 4, 6	Ape: Correzione di Errori	Debugging
Incontro 5	Corso 2 Lezione 13	2, 4, 6, 7	Ape: Istruzioni Condizionali	Introduzione blocco “se”
Incontro 6	Corso 3 Lezione 2	8, 9, 10	Labirinto: Sequenze e Cicli	Introduzione blocco “ripeti fino a che”
Incontro 7	Corso 3 Lezione 6	1, 2, 3	Ape: Funzioni	Definizione e richiamo di funzioni
Incontro 8	Corso 3 Lezione 7	1, 4, 5	Ape: Istruzioni Condizionali	Esercitazione con il blocco “se”

Dopo la fase di training per il gruppo sperimentale si è svolta la fase di post-test (T2) in cui, con le stesse modalità del pre-test, il gruppo sperimentale e quello di controllo hanno svolto i test di *coding* seguiti dal questionario di gradimento e, a distanza di qualche giorno, i test cognitivi ToL e Stroop Numerico. In questa fase il gruppo sperimentale ha svolto il questionario ESCAS, dopo aver acquisito dimestichezza con il termine *coding*. Questa seconda fase di test risulta cruciale per la dimostrazione dell’efficacia del trattamento. Ha reso infatti possibile rilevare il miglioramento delle abilità di *coding* tra

T1 e T2 e il confronto dei risultati ottenuti dal gruppo sperimentale (sottoposto a trattamento) con quelli ottenuti dal gruppo di controllo (non sottoposto a trattamento).

Successivamente, 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.

Una volta conclusa la fase di training per il gruppo di controllo, si è svolta l'ultima fase di post-test (T3) in cui entrambi i gruppi sono stati nuovamente sottoposti al test delle abilità di *coding*, seguito dal questionario di gradimento. In questa fase il gruppo di controllo è stato sottoposto al questionario ESCAS. In questo modo è stato possibile verificare l'efficacia del trattamento anche per il gruppo di controllo e stabilire se il gruppo sperimentale ha mantenuto nel tempo le abilità acquisite durante il training.

## Capitolo 5. Risultati

### 5.1 Analisi dei dati

#### 5.1.1 Analisi preliminari

Le analisi preliminari hanno permesso il confronto tra i gruppi per verificare che non ci fossero differenze statisticamente significative per età, status socioeconomico e familiarità con la tecnologia. Lo stesso confronto è stato fatto per i punteggi ottenuti nei test cognitivi Stroop e Torre di Londra a T1 in termini di accuratezza e per le prestazioni nel test di *coding* a T1.

Il test di Shapiro-Wilk ha mostrato che i dati relativi alle variabili considerate non seguono una distribuzione normale. Per questo, i confronti tra gruppi sono stati svolti utilizzando il test non parametrico di Wilcoxon-Mann-Whitney.

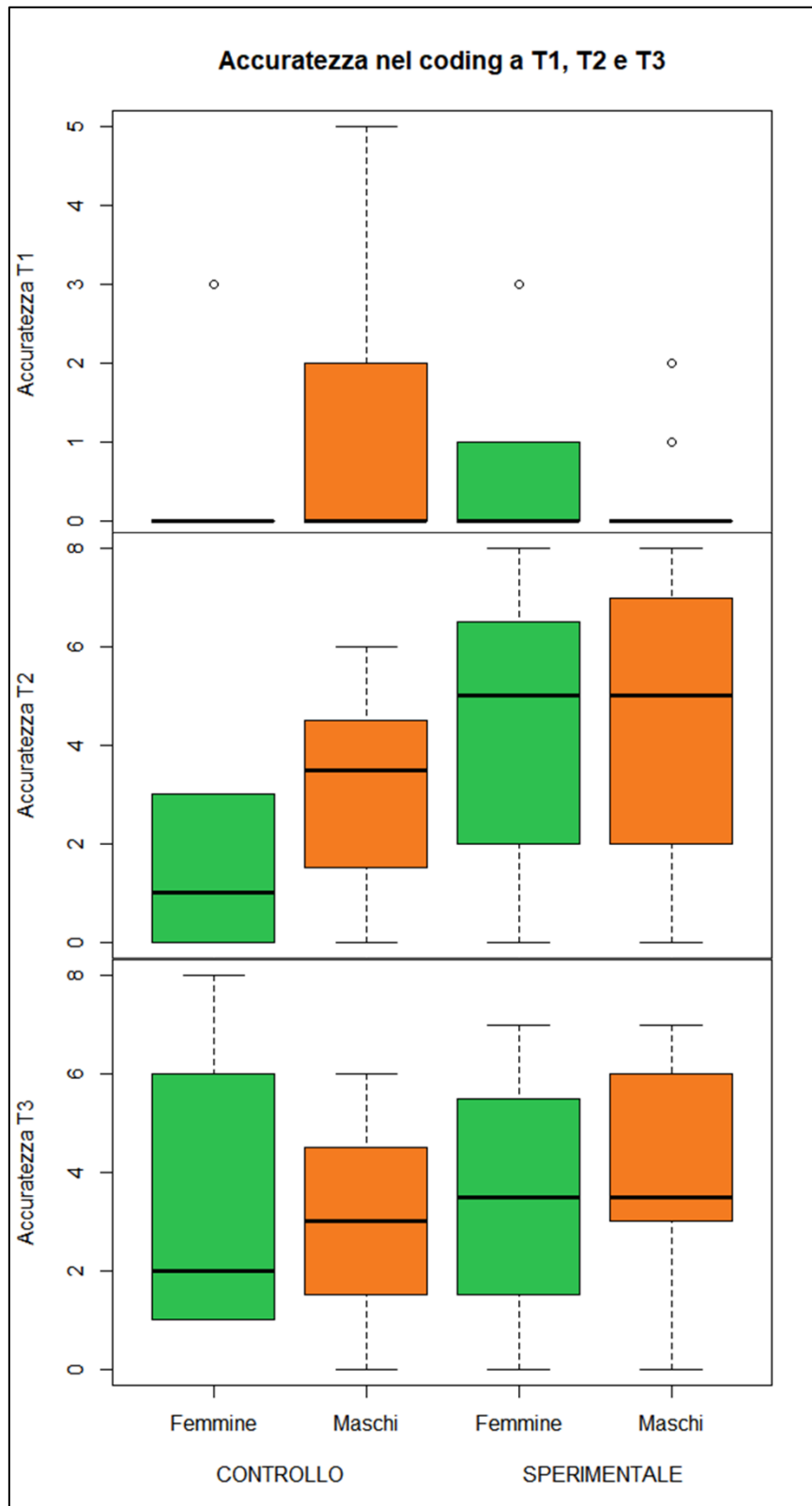
I risultati hanno dimostrato che, per tutte le variabili considerate, i due gruppi (sperimentale e controllo) erano omogenei tra loro:

- Età:  $W = 267; p = .386$
- SES:  $W = 203.5; p = .873$
- Fam Tech:  $W = 269; p = .324$
- Torre di Londra:  $W = 204.5; p = .525$
- Stroop:  $W = 255.5; p = .553$
- Coding:  $W = 251.5; p = .526$

È stato fatto lo stesso confronto anche all'interno dei gruppi per genere. I risultati sono mostrati nella **Tabella 1**. Anche in questo caso, non sono state riscontrate differenze statisticamente significative.

Il **Grafico 1** mostra i punteggi di accuratezza nel *coding* a T1, T2 e T3 per genere all'interno dei gruppi. A T1, nonostante l'analisi statistica riporti omogeneità tra i gruppi, l'analisi visiva del grafico suggerisce una diversa tendenza tra maschi e femmine. Questo

risultato potrebbe essere dovuto alla bassa numerosità campionaria e alla dimensione dell'effetto che risulta piccola per entrambi i confronti (Cohen's  $d = .321$  per il gruppo di controllo; Cohen's  $d = .258$  per il gruppo sperimentale).



**Grafico 1.** *Punteggi di accuratezza al coding a T1, T2 e T3.*

I due gruppi, quindi, possono essere considerati omogenei tra loro, non avendo trovato alcuna differenza statisticamente significativa nelle variabili prese in oggetto a T1.

### 5.1.2 Valutazione dell'intervento di training sulle abilità di *coding*

Successivamente, sono state eseguite le analisi di verifica di efficacia del training sulle abilità di *coding*. Nella **Tabella 3** sono riportati i punteggi di accuratezza al *coding* nei tre tempi. La rappresentazione visiva è mostrata nel **Grafico 1**.

**Tabella 3.**

*Medie e deviazioni standard dei punteggi di accuratezza nel coding a T1, T2 e T3, divisi per gruppo e per genere all'interno dei gruppi.*

Fase	Gruppo di controllo		Gruppo sperimentale	
	<i>M</i>	<i>SD</i>	<i>M</i>	<i>SD</i>
T1	0.81	1.54	0.36	0.79
T2	2.29	1.9	4.59	2.72
T3	3.14	2.29	3.77	2.16

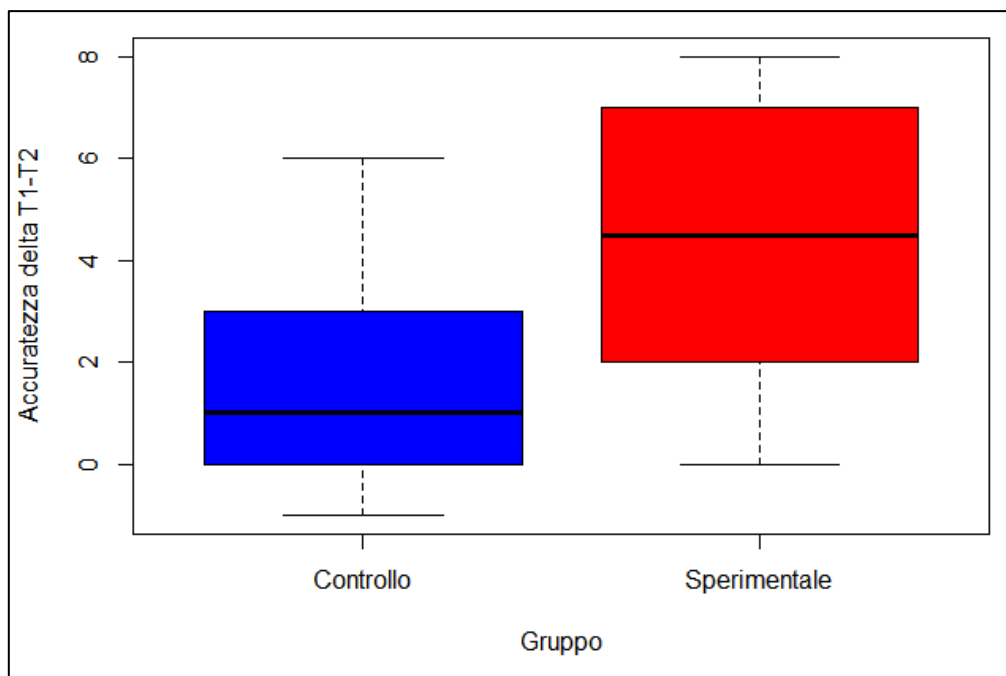
  

Fase	Femmine		Maschi		Femmine		Maschi	
	<i>M</i>	<i>SD</i>	<i>M</i>	<i>SD</i>	<i>M</i>	<i>SD</i>	<i>M</i>	<i>SD</i>
T1	0.33	1	1.17	1.8	0.63	1.06	0.21	0.58
T2	1.33	1.32	3	2	4.38	2.88	4.71	2.73
T3	3.44	2.79	2.92	1.93	3.5	2.45	3.93	2.06

Per confrontare il guadagno ottenuto nei punteggi di accuratezza al *coding* sono state create *ad hoc* due variabili di miglioramento, calcolando rispettivamente il delta tra T1 e T2 e il delta tra T2 e T3, ovvero la differenza tra i punteggi nelle fasi sperimentali. Il confronto è stato svolto tramite il test di Wilcoxon-Mann-Whitney, data la non normalità di distribuzione dei dati per una delle variabili.

Per la variabile di miglioramento T1-T2, i risultati del test di Wilcoxon-Mann-Whitney hanno mostrato una differenza statisticamente significativa tra i due gruppi, a vantaggio del gruppo sperimentale:  $W = 93$ ,  $p = .001$ , Cohen's  $d = .516$  (media). Questo rende evidente l'efficacia dell'intervento sul gruppo sperimentale che a seguito del training di *coding* ha ottenuto un miglioramento significativamente maggiore nei punteggi di accuratezza rispetto al gruppo di controllo che in questa fase ha proseguito le regolari attività didattiche.

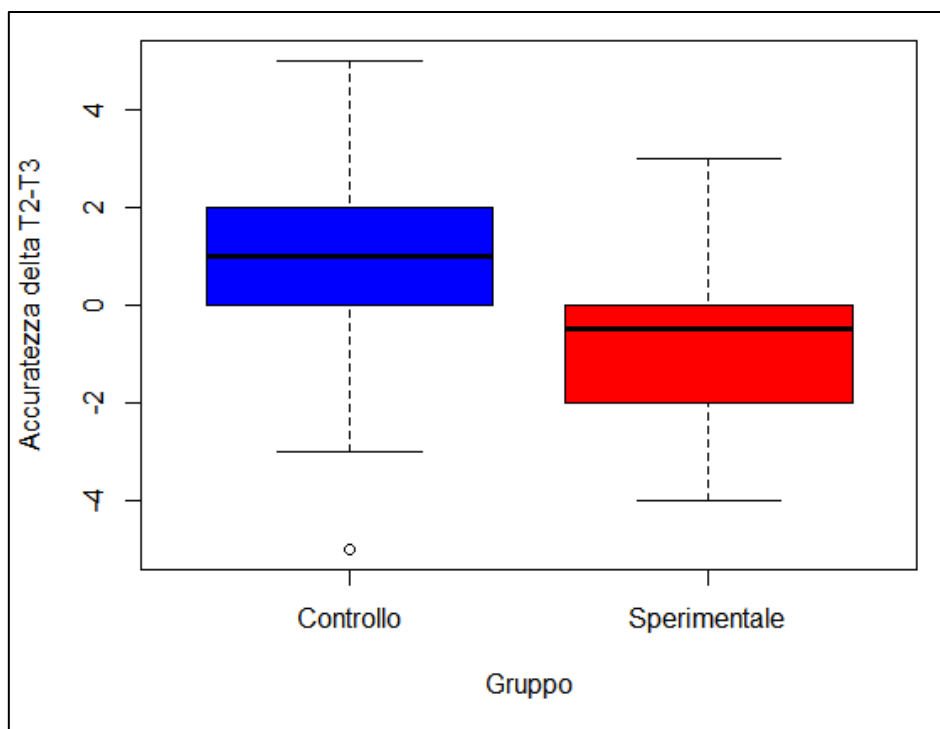
La differenza di miglioramento tra i due gruppi è mostrata dal **Grafico 2**.



**Grafico 2.** Miglioramento T1-T2 nell'accuratezza al coding.

Successivamente, è stata svolta la stessa analisi per la variabile di miglioramento T2-T3. Tra queste due fasi il gruppo di controllo, di tipo *waiting list*, è stato sottoposto all'intervento di training. I risultati del test di Wilcoxon-Mann-Whitney hanno mostrato una differenza statisticamente significativa tra i punteggi ottenuti dai due gruppi, a vantaggio del gruppo di controllo:  $W = 327$ ,  $p = .01$ , Cohen's  $d = .359$  (piccola). Anche per il gruppo di controllo si può quindi affermare che l'intervento è stato efficace, avendo ottenuto, dopo il training, un miglioramento significativamente maggiore nei punteggi di

accuratezza rispetto al gruppo sperimentale, che tra T2 e T3, periodo in cui ha svolto le regolari attività didattiche, non ha mostrato miglioramenti. Il delta T2-T3 è mostrato nel **Grafico 3**.



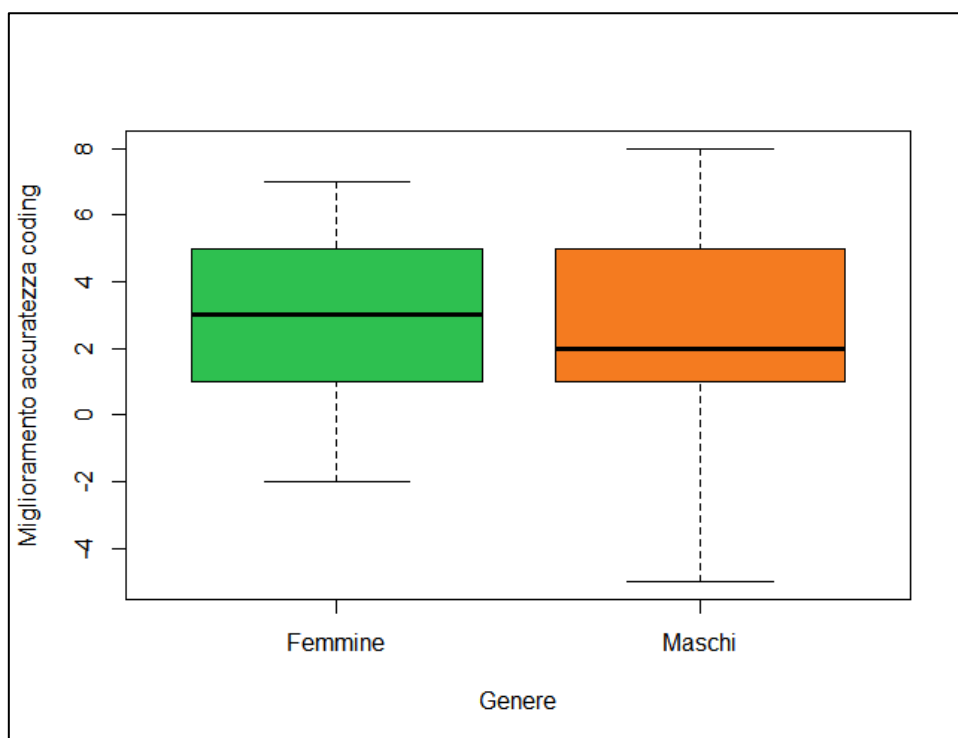
**Grafico 3.** *Miglioramento T2-T3 nell'accuratezza al coding.*

Questi risultati confermano l'ipotesi 1 dello studio, mostrando l'effetto positivo del training sul miglioramento nell'accuratezza al *coding* per entrambi i gruppi. Tuttavia, si può notare una differenza rispetto alla dimensione dell'effetto: per il confronto tra il delta T1-T2 la dimensione dell'effetto è media (Cohen's  $d = .516$ ), mentre per il delta T2-T3 è piccola (Cohen's  $d = .359$ ). Seppur non riscontrato dall'analisi statistica, l'analisi visiva del **Grafico 1** e le medie riportate nella **Tabella 3** mostrano a T1 livelli di accuratezza al *coding* maggiori per il gruppo di controllo, nello specifico per i maschi. Inoltre, tra T1 e T2 il gruppo di controllo ha mostrato dei miglioramenti, pur non avendo essendo stato sottoposto all'intervento. Di conseguenza, l'effetto di miglioramento tra T2 e T3 nel gruppo di controllo risulta piccolo se paragonato all'effetto mostrato dal gruppo sperimentale tra T1 e T2.

### 5.1.3 Differenze di genere nel miglioramento delle abilità di *coding*

È stata poi analizzata la possibile differenza di genere nell'effetto del training sul miglioramento delle abilità di *coding*. Anche in questo caso è stata creata *ad hoc* una variabile che verrà chiamata "efficacia" che considera per ogni soggetto il miglioramento ottenuto tra le fasi precedenti e successive al training, ovvero T1-T2 per i partecipanti del gruppo sperimentale e T2-T3 per quelli del gruppo di controllo. La variabile tiene quindi conto del gruppo di appartenenza dei soggetti. Questo ha permesso di fare un unico confronto per genere.

Il test di Wilcoxon-Mann-Whitney ha mostrato una differenza non significativa tra i due gruppi:  $W = 240$ ,  $p = .69$  (vedi **Grafico 4**). Questo risultato suggerisce che l'effetto del training sia lo stesso per maschi e femmine. Di conseguenza, l'ipotesi 2 di un maggior miglioramento di accuratezza a seguito del training per i maschi non è stata confermata.



**Grafico 4.** *Miglioramento nell'accuratezza al coding a seguito del training, confronto per genere.*



#### 5.1.4 Relazione tra autoefficacia e abilità di *coding*

Le analisi successive si sono concentrate sull'autoefficacia, in particolare sui punteggi ottenuti nelle scale ASCP (inclusa la sottoscala STEM) ed ESCAS (incluse le sottoscale di interesse, utilità percepita e confidence).

In primo luogo, è stata indagata la correlazione tra la scala ASCP, somministrata a T1, e la scala ESCAS, somministrata dopo l'intervento, ovvero a T2 per il gruppo sperimentale e a T3 per il gruppo di controllo. Questa analisi di tipo esplorativo vuole rispondere a questa domanda: l'autoefficacia legata al *coding* correla con l'autoefficacia scolastica e in particolare con l'autoefficacia STEM? La **Tabella 4** mostra una correlazione significativa tra la scala ASCP totale e la sottoscala ESCAS di utilità percepita ( $r = .416, p = .006$ ). Non sono state trovate altre correlazioni significative tra la scala ASCP e la scala ESCAS.

**Tabella 4.** Correlazioni tra la scala ASCP (totale e STEM) e la scala ESCAS (totale, interesse, confidence e utilità percepita).

	ASCP				ESCAS							
	Totale		STEM		Totale		Interesse		Confidence		Utilità percepita	
	<i>r</i>	<i>p</i>	<i>r</i>	<i>p</i>	<i>r</i>	<i>p</i>	<i>r</i>	<i>p</i>	<i>r</i>	<i>p</i>	<i>r</i>	<i>p</i>
ASCP TOTALE	1	<.0001***	.702	<.0001***	.279	.074	.168	.288	.126	.428	.416	<b>.006**</b>
ASCP STEM	.702	<.0001***	1	<.0001***	.108	.496	.07	.662	.071	.655	.126	.428
ESCAS TOTALE	.279	.074	.108	.496	1	<.0001***	.789	<.0001***	.835	<.0001***	.748	<.0001***
ESCAS (interesse)	.168	.288	.07	.662	.789	<.0001***	1	<.0001***	.456	.002**	.431	.004**
ESCAS (confidence)	.126	.428	.071	.655	.835	<.0001***	.456	.002**	1	<.0001***	.439	.003**
ESCAS (utilità percepita)	.416	<b>.006*</b>	.126	.428	.748	<.0001***	.431	.004**	.439	.003**	1	<.0001***

*Nota.* \* =  $p < .05$ ; \*\* =  $p < .01$ ; \*\*\* =  $p < .001$ .

Successivamente, è stata indagata la correlazione tra le due scale anche separatamente per genere, come mostrato nella **Tabella 5**. In questo caso, per le femmine non è stata riscontrata una correlazione tra le diverse scale, mentre per i maschi è presente una correlazione tra la scala ASCP totale e la sottoscala ESCAS di utilità percepita ( $r = 0.467, p = 0.019$ ).

**Tabella 5.** Correlazioni tra la scala ASCP (totale e STEM) e la scala ESCAS (totale, interesse, utilità percepita e confidence) per genere.

	ASCP				ESCAS							
	Totale		STEM		Totale		Interesse		Confidence		Utilità percepita	
	<i>r</i>	<i>p</i>	<i>r</i>	<i>p</i>	<i>r</i>	<i>p</i>	<i>r</i>	<i>p</i>	<i>r</i>	<i>p</i>	<i>r</i>	<i>p</i>
<b>Femmine</b>												
ASCP TOTALE	1	<.0001***	.802	.0001***	.035	.895	-.042	.874	-.054	.839	.208	.424
ASCP STEM	.802	.0001***	1	<.0001***	.29	.259	.256	.322	.284	.269	.118	.651
ESCAS TOTALE	.035	.895	.29	.259	1	<.0001***	.775	.0003***	.834	<.0001***	.717	.001*
ESCAS (interesse)	-.042	.874	.256	.322	.775	.0003***	1	<.0001***	.468	.058	.382	.13
ESCAS (confidence)	-.054	.839	.284	.269	.834	<.0001***	.468	.058	1	<.0001***	.371	.142
ESCAS (utilità percepita)	.208	.424	.118	.651	.717	.001**	.382	.13	.371	.142	1	<.0001*
<b>Maschi</b>												
ASCP TOTALE	1	<.0001***	.601	.002**	.329	.108	.246	.237	.13	.535	.467	<b>.019*</b>
ASCP STEM	.601	.002**	1	<.0001***	-.152	.468	-.071	.736	-.244	.24	-.01	.964
ESCAS TOTALE	.329	.108	-.152	.468	1	<.0001***	.801	<.0001***	.824	<.0001***	.745	<.0001***
ESCAS (interesse)	.246	.237	-.071	.736	.801	<.0001***	1	<.0001***	.447	.022*	.448	.022*
ESCAS (confidence)	.13	.535	-.244	.24	.824	<.0001***	.447	.022*	1	<.0001***	.43	.028*
ESCAS (utilità percepita)	.467	<b>.019*</b>	-.01	.964	.745	<.0001***	.448	.022*	.43	.028*	1	<.0001***

Nota. \* =  $p < .05$ ; \*\* =  $p < .01$ ; \*\*\* =  $p < .001$ .

Sono stati poi confrontati i punteggi ottenuti nella scala ASCP (totale e STEM) e nella scala ESCAS (totale e sotto scale) da maschi e femmine, indagando possibili differenze di genere. La **Tabella 6** riassume i risultati ottenuti tramite test di Wilcoxon-Mann-Whitney, data la non normalità di distribuzione di alcune delle variabili. È stata riscontrata una differenza di genere statisticamente significativa nei punteggi della sottoscala ASCP STEM, a vantaggio dei maschi:  $W = 128$ ,  $p = .028$ , Cohen's  $d = .341$  (piccolo) (**Grafico 5**). Per i punteggi della scala ASCP totale e della scala ESCAS totale e relative sottoscale non sono state ritrovate differenze di genere significative.

**Tabella 6.** *Confronto per genere tra i punteggi ottenuti nella scala ASCP (totale e STEM) e nella scala ESCAS (totale, interesse, utilità percepita e confidence).*

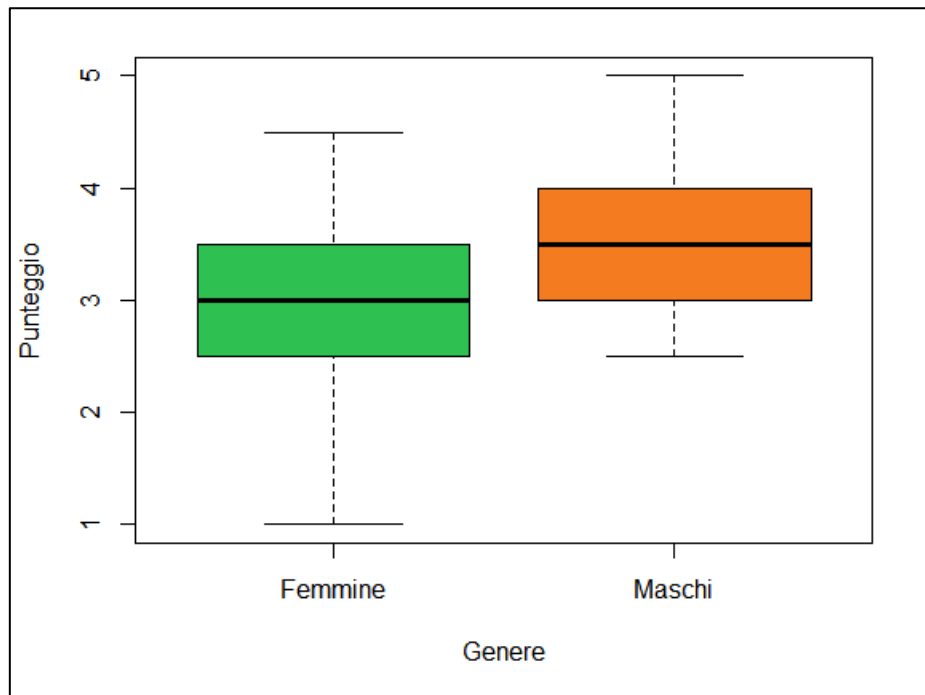
	Genere	
	<i>W</i>	<i>p - value</i>
ASCP TOTALE	143	.075
ASCP STEM	128	<b>.028 *</b>
ESCAS TOTALE	158	.12
ESCAS (interesse)	166	.172
ESCAS (confidence)	168.5	.196
ESCAS (utilità percepita)	158	.118

*Nota.* \* =  $p < .05$ ; \*\* =  $p < .01$ ; \*\*\* =  $p < .001$ .

L'ipotesi 3, ovvero la presenza di una differenza di genere nei punteggi di autoefficacia STEM e autoefficacia legata al *coding* a seguito del training, è stata quindi confermata solo in parte.

È interessante osservare che il livello di autoefficacia scolastica, misurato prima dell'inizio del training, era uguale tra maschi e femmine, fatta eccezione per l'autoefficacia relativa alle discipline STEM, in cui le femmine hanno riportato un livello minore. Nonostante questa differenza, a seguito del training maschi e femmine hanno

riportato gli stessi livelli di interesse, autoefficacia e utilità percepita rispetto alla programmazione. Questo potrebbe essere dovuto all'intervento di training o ad altre cause, come ad esempio l'assenza di correlazione significativa tra le scale ASCP ed ESCAS per le femmine e limitata ad ASCP totale ed ESCAS utilità percepita per i maschi, ma non è possibile affermarlo con certezza.



**Grafico 5.** *Punteggio scala ASCP STEM, confronto per genere.*

Infine, è stata presa in analisi la possibile correlazione tra la variabile di efficacia del trattamento e le scale ASCP ed ESCAS. I risultati non hanno mostrato alcuna correlazione significativa (**Tabella 7**).

**Tabella 7.** Correlazioni tra la scala ASCP (totale e STEM) e la scala ESCAS (totale, interesse, utilità percepita e confidence) con la variabile di efficacia.

	<b>Efficacia</b>	
	<i>r</i>	<i>p - value</i>
ASCP TOTALE	.18	.25
ASCP STEM	.13	.4
ESCAS TOTALE	- .05	.74
ESCAS (interesse)	.17	.28
ESCAS (confidence)	- .21	.18
ESCAS (utilità percepita)	- .06	.72

*Nota.* \* =  $p < .05$ ; \*\* =  $p < .01$ ; \*\*\* =  $p < .001$ .

Di conseguenza, l'ipotesi 4 non è stata confermata: maggiori punteggi di autoefficacia STEM, misurati a T1, non hanno predetto maggiori miglioramenti nel *coding* a seguito del training; allo stesso tempo, maggiori miglioramenti ottenuti dopo l'intervento non hanno predetto un maggiore livello di autoefficacia legata al *coding*, misurata nella fase successiva al trattamento tramite la scala ESCAS.

## 5.2 Discussione e conclusioni

Questo studio è stato svolto con lo scopo di verificare l'efficacia di un training di *coding* su studenti di classe quarta della scuola primaria e di indagare la presenza di differenze di genere in questo effetto. Inoltre, è stato esaminato il ruolo dell'autoefficacia, in particolare quella legata al *coding*, sul miglioramento nelle abilità di *coding*.

Coerentemente con quanto atteso, l'intervento di training sul *coding* si è rivelato essere efficace. Sia il gruppo *waiting list* che il gruppo sperimentale hanno ottenuto un miglioramento nei punteggi di accuratezze al *coding* a seguito del trattamento. Questi risultati sono in accordo con quanto emerso da studi analoghi precedenti (Arfé et al., 2019, 2020; Montuori et al., 2022).

L'efficacia del trattamento è risultata essere pari sia per i maschi che per le femmine. Non sono state infatti riscontrate differenze di genere statisticamente significative nel miglioramento nei punteggi di accuratezza al *coding*. Per quanto questo risultato non confermi l'ipotesi di un maggiore miglioramento per i maschi, è in linea con altri studi in fascia d'età 9-11 anni (Jiang & Wong, 2021; Kožuh et al., 2018; Price & Price-Mohr, 2023) Anche in questi studi non è stata riscontrata una differenza di genere nei compiti di *coding*. Tuttavia, i dati in letteratura non sono sempre concordanti, sebbene non risulti chiara la direzione dell'effetto di genere: Román-González et al. (2017) hanno riportato un vantaggio per i maschi, mentre Statter e Armoni (2017) per le femmine. Risulta evidente che questo fenomeno debba continuare ad essere approfondito tramite ulteriori ricerche.

Inoltre, il risultato in merito alla differenza di genere non si allinea ai risultati riportati dallo studio analogo di riferimento di Montuori et al. (2022) su un campione di alunni italiani, in cui è stata riscontrata, solamente nella fase T2 successiva al training, una differenza di genere significativa nell'accuratezza al *coding*, a vantaggio dei maschi. Una spiegazione di questa discrepanza potrebbe essere legata alla dimensione dell'effetto e alla numerosità campionaria. Nello studio di riferimento, a cui hanno preso parte 109 bambini, il confronto per genere tra i punteggi di accuratezza al *coding* a T2 è risultato statisticamente significativo con dimensione dell'effetto media (Cohen's  $d = - 0.41$ ). Nel presente studio, a cui hanno preso parte un numero minore di soggetti (43), la dimensione dell'effetto in merito alla differenza di genere nel miglioramento al *coding* è piccola (Cohen's  $d = - 0.07$ ). La differenza di genere in questo confronto potrebbe non essere stata riscontrata proprio a causa di un campione non sufficientemente numeroso per rilevare una differenza di genere nella risposta all'intervento.



Rispetto ai livelli di autoefficacia, l'ipotesi di differenza di genere è stata parzialmente confermata. Prima del training, i maschi hanno riportato maggiori livelli di autoefficacia in ambito STEM, tuttavia il livello generale di autoefficacia scolastica percepita è risultato pari per genere. A seguito del training, invece, non sono state riscontrate differenze di genere nei livelli di autoefficacia legata al *coding*, in accordo con quanto riportato da Kong et al. (2018), ma diversamente rispetto alle evidenze di Tellhed et al. (2022).

Il risultato riscontrato in questo studio potrebbe suggerire che in studenti di classe quarta della scuola primaria un intervento di training sul *coding* mostra avere buoni esiti sui livelli di interesse e autoefficacia legata al *coding*, potenzialmente analoghi in maschi e femmine. Tuttavia, non possiamo affermare con certezza che l'intervento abbia avuto un effetto correttivo sulle differenze di genere mostrate nei livelli di autoefficacia.

Infine, i maggiori livelli di autoefficacia nelle discipline STEM da parte dei maschi non sono risultati essere correlati a un maggiore miglioramento dei punteggi di accuratezza al *coding*, non confermando la quarta e ultima ipotesi dello studio. Questo potrebbe significare che l'autoefficacia non svolge un ruolo significativo sullo sviluppo delle abilità di *coding* e in generale sullo sviluppo del pensiero computazionale. Tuttavia, questa affermazione risulterebbe fortemente in contrasto con le evidenze presenti in letteratura. Tipicamente, maggiori livelli di autoefficacia portano a un maggiore impegno verso il compito, aumentando le probabilità di successo (Bandura, 2000). Kong et al. (2018) ritengono l'autoefficacia legata al *coding* come uno degli elementi fondamentali per lo sviluppo delle abilità di programmazione. Inoltre, è dimostrato che lo sviluppo delle abilità di *coding*, in base al contesto di apprendimento e alle attività proposte, può favorire il senso di autoefficacia (Fletcher et al., 2021; Güdel et al., 2019; Leonard et al., 2016; Statter & Armoni, 2017). Una spiegazione più plausibile di questo risultato è la non

sufficiente dimensione del campione oppure l'influenza di altri fattori, come la presenza di stereotipi, il contesto culturale e di apprendimento o la modalità e il tipo di attività proposte. Inoltre, l'intervento è stato svolto da parte di modelli femminili e questo potrebbe aver contrastato l'effetto delle percezioni di efficacia sulla prestazione.

### **5.2.1 Limiti e sviluppi futuri**

Uno dei principali limiti di questo studio è la dimensione del campione, probabilmente non sufficiente da mostrare significatività statistica in alcune analisi svolte, ipotesi supportata da dimensioni dell'effetto in generale piccole. Studi futuri dovrebbero essere svolti con campioni più ampi, coinvolgendo più classi di pari grado di istruzione. Questo potrebbe portare a dimensioni dell'effetto maggiori, dando maggiore solidità ai risultati trovati.

Merita inoltre fare una riflessione sui punteggi di accuratezza al *coding* nelle tre fasi sperimentali, rappresentati nel **Grafico 1**. Tra il gruppo di controllo e il gruppo sperimentale, che rappresentano due classi diverse dello stesso istituto scolastico, è possibile notare a livello visivo pattern di andamento differenti, anche se non riscontrati a livello statistico. In particolare, nel gruppo di controllo i maschi avevano un livello di partenza maggiore rispetto alle femmine, tendenza che si è mantenuta a T2. A seguito dell'intervento, il punteggio medio di accuratezza (**Tabella 3**) è simile tra maschi e femmine. Tuttavia, le femmine mostrano maggiore dispersione dei punteggi, tendenti verso il basso, come mostrato dalla mediana, inferiore rispetto ai maschi. Al contrario, nel gruppo sperimentale, rispetto ai maschi, le femmine avevano un livello di partenza leggermente superiore e prestazioni simili a seguito dell'intervento. Nell'ultima fase sperimentale, svolta qualche settimana dopo T2, il gruppo sperimentale ha mantenuto prestazioni medie simili tra maschi e femmine, ma le femmine hanno mostrato maggiore dispersione verso punteggi più bassi, con mediana pari a quella dei

maschi. Questo diverso andamento potrebbe essere dovuto a fattori legati al gruppo classe, come il contesto di apprendimento, il rapporto tra pari e le modalità di insegnamento delle docenti. L'effetto dovuto al contesto scolastico e in particolare alle dinamiche del gruppo classe sull'autoefficacia e sulle prestazioni nelle attività di pensiero computazionale dovrebbe essere approfondito da ulteriori ricerche.

Infine, altri fattori da approfondire sono la cultura di appartenenza e lo status socioeconomico. Lo studio è infatti stato svolto in un quartiere di Padova con alta presenza migratoria. Pertanto, anche il campione di riferimento mostra diversi contesti culturali, oltre che un generale livello di SES basso, non rappresentativo della popolazione. Ci sono evidenze che la credenza rispetto alla programmazione come compito tipicamente svolto da maschi sia un fenomeno culturalmente locato (Tellhed et al., 2022). Studi futuri dovrebbero dunque indagare la presenza di differenze di genere in questo ambito, con riferimento alle diverse provenienze culturali.

In conclusione, questo studio ha mostrato l'efficacia di un intervento di *coding* in alunni di classe quarta della scuola primaria, con effetti pari per maschi e femmine. Inoltre, nei livelli di autoefficacia, fatta eccezione per l'autoefficacia verso materie STEM, non sono state riscontrate differenze di genere. Infine, il livello di autoefficacia non è sembrato essere correlato ai miglioramenti ottenuti tramite l'intervento. Contribuire a studiare la presenza di differenze di genere nell'ambito del pensiero computazionale e il ruolo dell'autoefficacia in questo fenomeno risulta importante per giungere ad evidenze più solide rispetto a quelle attualmente presenti in letteratura, i cui risultati si sono mostrati talvolta essere in contrasto. Questo permetterebbe di capire al meglio le cause sottostanti al *gender gap* in ambito STEM e ad individuare possibili strategie da mettere in atto per incentivare bambine e ragazze a *programmare il futuro*.

## Bibliografia

- Aho, A. V. (2012). Computation and Computational Thinking. *The Computer Journal*, 55(7), 832–835. <https://doi.org/10.1093/comjnl/bxs074>
- Arfè, B., & Vardanega, T. (2019). Imparare a ragionare: Il ruolo del pensiero computazionale a scuola. *Giornale italiano di psicologia*, 4, 765–770. <https://doi.org/10.1421/95548>
- Arfé, 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>
- Arfé, 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>
- Baddeley, A. (2010). Working memory. *Current Biology*, 20(4), R136–R140.
- Bandura, A. (1986). *Social foundations of thought and action: A social cognitive theory / Albert Bandura*. Prentice-Hall.
- Bandura, A. (2000). *Autoefficacia: Teoria e applicazioni / Albert Bandura ; presentazione di Gian Vittorio Caprara ; traduzione di Gabriele Lo Iacono*. Erickson.
- Bandura, A., Barbaranelli, C., Caprara, G. V., & Pastorelli, C. (2001). Self-Efficacy Beliefs as Shapers of Children’s Aspirations and Career Trajectories. *Child Development*, 72(1), 187–206. <https://doi.org/10.1111/1467-8624.00273>
- Barth-Cohen, L., Montoya, B., & Shen, J. (2019). Making in the Middle: Walk Like a Robot. *Science Scope*, 042(09). [https://doi.org/10.2505/4/ss19\\_042\\_09\\_12](https://doi.org/10.2505/4/ss19_042_09_12)

- Bati, K. (2022). A systematic literature review regarding computational thinking and programming in early childhood education. *Education and Information Technologies*, 27(2), 2059–2082. <https://doi.org/10.1007/s10639-021-10700-2>
- Bell, T., Alexander, J., Freeman, I., & Grimley, M. (2009). Computer Science Unplugged: School students doing real computing without computers. *The New Zealand Journal of Applied Computing and Information Technology*, 13.
- Bergey, B. W., Ketelhut, D. J., Liang, S., Natarajan, U., & Karakus, M. (2015). Scientific Inquiry Self-Efficacy and Computer Game Self-Efficacy as Predictors and Outcomes of Middle School Boys' and Girls' Performance in a Science Assessment in a Virtual Environment. *Journal of Science Education and Technology*, 24(5), 696–708. <https://doi.org/10.1007/s10956-015-9558-4>
- Bocconi, S., Chiocciariello, A., Kampylis, P., Dagienė, V., Wastiau, P., Engelhardt, K., Earp, J., Horvath, M., Jasuté, E., Malagoli, C., Masiulionytė-Dagienė, V., & Stupurienė, G. (2022). *Reviewing computational thinking in compulsory education: State of play and practices from computing education*. Publications Office. <https://data.europa.eu/doi/10.2760/126955>
- Bouffard-Bouchard, T. (1990). Influence of Self-Efficacy on Performance in a Cognitive Task. *Journal of Social Psychology*, 130(3), 353–363. <https://doi.org/10.1080/00224545.1990.9924591>
- Bucciarelli, M. (2019). Imparare a ragionare... E continuare a farlo. *Giornale italiano di psicologia*, 4/2019. <https://doi.org/10.1421/95546>
- Busch, T. (1995). Gender Differences in Self-Efficacy and Attitudes toward Computers. *Journal of Educational Computing Research*, 12(2), 147–158. <https://doi.org/10.2190/H7E1-XMM7-GU9B-3HWR>

- Caprara, G. V. (2001). *La valutazione dell'autoefficacia: Costrutti e strumenti / Gian Vittorio Caprara (a cura di)*. Erickson.
- Cetin, I. (2016). Preservice Teachers' Introduction to Computing: Exploring Utilization of Scratch. *Journal of Educational Computing Research*, 54(7), 997–1021.  
<https://doi.org/10.1177/0735633116642774>
- Cornoldi, C., De Beni, R., Zamperlin, C., & Meneghetti, C. (2005). *AMOS 8-15. Abilità e motivazione allo studio: Prove di valutazione per ragazzi dagli 8 ai 15 anni*. Trento, Erickson.
- Dağ, F., Şumuer, E., & Durdu, L. (2023). The effect of an unplugged coding course on primary school students' improvement in their computational thinking skills. *Journal of Computer Assisted Learning*, jcal.12850. <https://doi.org/10.1111/jcal.12850>
- Di Lieto, M. C., Inguaggiato, E., Castro, E., Cecchi, F., Cioni, G., Dell'Omo, M., Laschi, C., Pecini, C., Santerini, G., Sgandurra, G., & Dario, P. (2017). Educational Robotics intervention on Executive Functions in preschool children: A pilot study. *Computers in Human Behavior*, 71, 16–23. <https://doi.org/10.1016/j.chb.2017.01.018>
- Fancello, G. S., Vio, C., & Cianchetti, C. (2006). *ToL, Torre di Londra. Test di Valutazione delle Funzioni Esecutive (Pianificazione e Problem Solving)*. Centro studi Erickson.
- Fletcher, A., Mason, R., & Cooper, G. (2021). Helping students get IT: Investigating the longitudinal impacts of IT school outreach in Australia. *Proceedings of the 23rd Australasian Computing Education Conference*, 115–124.  
<https://doi.org/10.1145/3441636.3442312>
- Flórez, F. B., 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>

Frymier, A. B., Shulman, G. M., & Houser, M. (1996). The development of a learner empowerment measure. *Communication Education*, 45(3), 181–199.

<https://doi.org/10.1080/03634529609379048>

Grover, S., Pea, R., & Cooper, S. (2015). Designing for deeper learning in a blended computer science course for middle school students. *Computer Science Education*, 25(2), 199–237. <https://doi.org/10.1080/08993408.2015.1033142>

Güdel, K., Heitzmann, A., & Müller, A. (2019). Self-efficacy and (vocational) interest in technology and design: An empirical study in seventh and eighth-grade classrooms. *International Journal of Technology and Design Education*, 29(5), 1053–1081.

<https://doi.org/10.1007/s10798-018-9475-y>

Jiang, S., & Wong, G. K. W. (2021). Exploring age and gender differences of computational thinkers in primary school: A developmental perspective. *Journal of Computer Assisted Learning*, 38(1), 60–75. <https://doi.org/10.1111/jcal.12591>

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>

Kim, M. H., Anderson, R. C., DeRosia, N., Madison, E., & Husman, J. (2021). There are two I's in motivation: Interpersonal dimensions of science self-efficacy among racially diverse adolescent youth. *Contemporary Educational Psychology*, 66, 101989.

<https://doi.org/10.1016/j.cedpsych.2021.101989>

Kong, S.-C., Chiu, M. M., & Lai, M. (2018). A study of primary school students' interest, collaboration attitude, and programming empowerment in computational thinking

education. *Computers & Education*, 127, 178–189.

<https://doi.org/10.1016/j.compedu.2018.08.026>

Kožuh, I., Krajnc, R., Hadjileontiadis, L. J., & Debevc, M. (2018). Assessment of problem solving ability in novice programmers. *PLOS ONE*, 13(9), e0201919.

<https://doi.org/10.1371/journal.pone.0201919>

Lee, J., & Junoh, J. (2019). Implementing Unplugged Coding Activities in Early Childhood Classrooms. *Early Childhood Education Journal*, 47(6), 709–716.

<https://doi.org/10.1007/s10643-019-00967-z>

Leonard, J., Buss, A., Gamboa, R., Mitchell, M., Fashola, O. S., Hubert, T., & Almughyirah, S. (2016). Using Robotics and Game Design to Enhance Children’s Self-Efficacy, STEM Attitudes, and Computational Thinking Skills. *Journal of Science Education and Technology*, 25(6), 860–876. <https://doi.org/10.1007/s10956-016-9628-2>

Lishinski, A., Yadav, A., Good, J., & Enbody, R. (2016). Learning to Program: Gender Differences and Interactive Effects of Students’ Motivation, Goals, and Self-Efficacy on Performance. *Proceedings of the 2016 ACM Conference on International Computing Education Research*, 211–220. <https://doi.org/10.1145/2960310.2960329>

Marzocchi, G. M., Cornoldi, C., & Re, A. M. (2010). *BIA, Batteria italiana per l’ADHD per la valutazione dei bambini con deficit di attenzione e iperattività*. Erickson.

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. (2021). Gender Stereotypes Influence Children’s STEM Motivation. *Child Development Perspectives*, 15(3), 203–210. <https://doi.org/10.1111/cdep.12424>



- 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>
- Mayall, H. (2008). Differences in gender based technology self-efficacy across academic levels. *International Journal of Instructional Media*, *35*(2), 145.
- Metin, S. (2022). Activity-based unplugged coding during the preschool period. *International Journal of Technology and Design Education*, *32*(1), 149–165. <https://doi.org/10.1007/s10798-020-09616-8>
- Miller, D. I., & Halpern, D. F. (2014). The new science of cognitive sex differences. *Trends in Cognitive Sciences*, *18*(1), 37–45. <https://doi.org/10.1016/j.tics.2013.10.011>
- Miyake, A., Friedman, N. P., Emerson, M. J., Witzki, A. H., Howerter, A., & Wager, T. D. (2000). The Unity and Diversity of Executive Functions and Their Contributions to Complex “Frontal Lobe” Tasks: A Latent Variable Analysis. *Cognitive Psychology*, *41*(1), 49–100. <https://doi.org/10.1006/cogp.1999.0734>
- Moè, A. (2020). *La motivazione Teorie e processi*. Società editrice il Mulino, Spa.
- 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. (2019). Do we really need computational thinking? *Communications of the ACM*, *62*(2), 32–35. <https://doi.org/10.1145/3231587>
- Nardelli, E., & Ventre, G. (2015). *Introducing Computational Thinking in Italian Schools: A First Report on “Programma Il Futuro” Project*.

- Nesbitt, K. T., Baker-Ward, L., & Willoughby, M. T. (2013). Executive function mediates socio-economic and racial differences in early academic achievement. *Early Childhood Research Quarterly, 28*(4), 774–783. <https://doi.org/10.1016/j.ecresq.2013.07.005>
- 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: Children, Computers, and Powerful Ideas*. New York: Basic Books, Inc., Publishers.
- Pastorelli, C., & Piccioni, L. (2001). Scala di Autoefficacia Scolastica Percepita. In G.V. Caprara (a cura di), *La valutazione dell'autoefficacia*. Trento: Erickson.
- Price, C. B., & Price-Mohr, R. (2023). Exploring gender differences in primary school computer programming classes: A study in an English state-funded urban school. *Education 3-13, 51*(2), 306–319. <https://doi.org/10.1080/03004279.2021.1971274>
- Riedel, M., Streit, A., Wolf, F., Lippert, T., & Kranzlmüller, D. (2008). Classification of Different Approaches for e-Science Applications in Next Generation Computing Infrastructures. *2008 IEEE Fourth International Conference on eScience*, 198–205. <https://doi.org/10.1109/eScience.2008.56>
- 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>

- Rotter, J. B. (1966). Generalized expectancies for internal versus external control of reinforcement. *Psychological Monographs: General and Applied*, 80(1), 1–28.  
<https://doi.org/10.1037/h0092976>
- 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>
- 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>
- Statter, D., & Armoni, M. (2017). Learning Abstraction in Computer Science: A Gender Perspective. *Proceedings of the 12th Workshop on Primary and Secondary Computing Education*, 5–14. <https://doi.org/10.1145/3137065.3137081>
- Sullivan, A., & Bers, M. U. (2013). Gender differences in kindergarteners’ robotics and programming achievement. *International Journal of Technology and Design Education*, 23(3), 691–702. <https://doi.org/10.1007/s10798-012-9210-z>
- 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, 145–165.  
<https://doi.org/10.28945/3547>
- Tawfik, A. A., Payne, L., & Olney, A. M. (2022). Scaffolding Computational Thinking Through Block Coding: A Learner Experience Design Study. *Technology, Knowledge and Learning*. <https://doi.org/10.1007/s10758-022-09636-4>

- Tellhed, U., Björklund, F., & Kallio Strand, 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>
- UNESCO. (2017). *Cracking the code: Girls' and women's education in science, technology, engineering and mathematics (STEM)*. UNESCO. <https://doi.org/10.54675/QYHK2407>
- 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(1), 119–140. <https://doi.org/10.1007/s10648-015-9355-x>
- Wierenga, L. M., Bos, M. G. N., Van Rossenberg, F., & Crone, E. A. (2019). Sex Effects on Development of Brain Structure and Executive Functions: Greater Variance than Mean Effects. *Journal of Cognitive Neuroscience*, 31(5), 730–753. [https://doi.org/10.1162/jocn\\_a\\_01375](https://doi.org/10.1162/jocn_a_01375)
- 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*.
- Yasar, O. (2018). A new perspective on computational thinking. *Communications of the ACM*, 61(7), 33–39. <https://doi.org/10.1145/3214354>

## Sitografia

<https://www.istat.it/it/archivio/276497> (ISTAT, Livelli di istruzione e ritorni occupazionali. Anno 2021)

<http://programmailfuturo.it>

<https://code.org/>