



UNIVERSITÀ
DEGLI STUDI
FIRENZE

Scuola di
Studi Umanistici
e della Formazione

Corso di Laurea in
Scienze della Formazione Primaria N.O.

Programmare con Logo e Scratch

Gli effetti del coding sul concetto di angolo.

Relatore

Prof. Andreas Robert Formiconi

Candidato

Fabiola Izzo

Ringraziamenti

A mia sorella Cristina, per esserci sempre stata. Mi hai accettata per come sono, sostenendomi e stimolandomi sempre a migliorare. Sapere di avere qualcuno con cui affrontare le difficoltà è stato fondamentale, hai sempre creduto in me e nel mio percorso e tante volte sei stata il motivo per il quale non mi sono arresa.

Ai miei genitori, grazie per tutti i sacrifici che avete fatto per non farmi mancare mai niente, grazie per aver sempre creduto in me e nelle mie capacità, grazie per avermi sorretta nei momenti difficili, grazie per avermi mostrato le possibili strade senza indirizzarmi verso quella che per voi era giusta, mi avete aiutata a trovare da sola il mio cammino.

A Federica, sei diventata una sorella, sei sempre stata presente nei momenti più importanti della mia vita, riesci a sorridere anche nei giorni più bui e a contagiarmi con il tuo sorriso. Grazie per le parole di coraggio, per le ore passate ad ascoltare i miei sfoghi e per il sostegno che mi hai sempre dato.

A nonna Angela, sei stata l'esempio più importante della mia vita, mi hai mostrato come una donna possa cavarsela da sola, mi hai insegnato ad essere forte e che l'amore non si mostra in un'unica forma. I tuoi insegnamenti sono sempre con me.

A nonno Francesco e nonna Elena, il vostro amore per la famiglia e i vostri sacrifici hanno permesso a tutti noi di essere quelli che siamo oggi. Mi avete mostrato che con il duro lavoro si può raggiungere qualsiasi obiettivo.

A Sandra, siamo lontane ma sento vicini il tuo sostegno e il tuo supporto, tu, Tony, Simone e Alice siete diventati parte della nostra famiglia.

A Linda, che tante volte mi ha capita più di quanto io non abbia fatto con me stessa, sei un sostegno importante nella mia vita.

A Giulia e Tiziana, poter svolgere questo percorso con voi è stato fondamentale, abbiamo condiviso ansie e soddisfazioni, ci siamo sempre sostenute e stimolate. Tutte le ore passate a preparare progetti insieme e tutti i viaggi per andare all'università rimarranno un ricordo indelebile nella mia mente.

A Silvia ed Elena, anche se solo negli ultimi anni i viaggi e i corsi ci hanno unite, siete state un sostegno importante per la chiusura di questo percorso.

Ad Alessia, grazie per avermi aiutata a trovare me stessa e la forza di affrontare questo percorso e a Martina, Francesco, Cristiano, Paola e Giulio, che mi hanno fatta sentire capita e accettata.

Alle mie tutor scolastiche Rossana e Arianna, che mi hanno trasmesso la passione per questa professione e mi hanno dato fiducia lasciandomi mettere alla prova. Grazie per il vostro sostegno e per i consigli.

A Martina, poter condividere il progetto della tesi è stato stimolante e un sostegno nei momenti più difficili.

Ai bambini che hanno fatto parte del mio percorso, i vostri sorrisi e i vostri successi sono stati lo stimolo più grande.

Alla mia tutor universitaria Lucia Maddii, che in questi ultimi due anni mi ha insegnato molto ed è stata un esempio di come deve essere una maestra.

Un ringraziamento speciale va al Professor Andreas Robert Formiconi, un insegnante competente e stimolante, che ha creduto in questo progetto e mi ha trasmesso la sua passione e il suo entusiasmo.

Indice

Introduzione	9
Capitolo 1 - Coding e pensiero computazionale in ottica educativa	13
1.1 Coding e pensiero computazionale: una definizione	13
1.2 Nativi digitali	16
1.3 Presupposti teorici del coding e del pensiero computazionale: Seymour Papert.....	18
1.3.1 Piaget e il costruttivismo	18
1.3.2 Dewey e l'attivismo	22
1.3.3 Papert e Minsky.....	23
1.5 Il coding e le competenze chiave di cittadinanza.....	26
1.6 Il coding è una competenza digitale?.....	29
1.7 Perché fare il coding a scuola?.....	32
1.7.1 Insegnanti e coding	33
1.8 Proposte per lo sviluppo del pensiero computazionale nella realtà internazionale	34
1.8.1 Code.org	35
1.8.2 Hour of code.....	37
1.8.3 La sezione “studenti”	38
1.8.4 CoderDojo	39
1.9 La storia del pensiero computazionale nella scuola italiana	41
1.9.1 Piano nazionale scuola digitale	42
1.9.2 PON.....	44
1.10 Proposte per lo sviluppo del pensiero computazionale nella realtà italiana	45
1.10.1 Programma il futuro	45

1.10.2 CoderDojo in Italia.....	46
1.10.2 CodeMOOC	46
1.10.4 Logo	47
1.10.5 Scratch.....	48
1.10. 6 Robot educativi	49
1.10.7 Coding unplugged	52
Capitolo 2 - I linguaggi di programmazione	55
2.1 Cosa sono i linguaggi di programmazione?	55
2.2 Bricolage uno stile di apprendimento adatto alla programmazione.....	59
2.3 L'errore nella programmazione: un momento di apprendimento.....	61
2.4 Micromondi: uno spazio dove esplorare i linguaggi di programmazione...	64
2.5 Il linguaggio di Logo.....	64
2.6 Il linguaggio di Scratch	66
2.7 Coding testuale e visuale a confronto	67
2.8 I linguaggi di programmazione come supporto nell'apprendimento dei concetti geometrici	69
Capitolo 3 – Il progetto svolto in classe: coding con Logo e Scratch.....	73
3.1 La nascita del progetto	73
3.2 Descrizione del progetto	74
3.3 I soggetti della sperimentazione.....	75
3.4 Descrizione del progetto	75
3.5 Descrizione dell'esperienza	79
3.6 Presentazione dei test di geometria	91
3.7 Risultati del test.....	92
3.8 Presentazione del questionario	111
3.9 Analisi dei risultati del questionario.....	118

Conclusioni	129
Bibliografia	133
Sitografia	138
Appendice	140
a)Test 1	142
b)Test 2.....	147
c)Test 3.....	152
d) Librino con i comandi scritti dai bambini e alcune delle loro creazioni con Logo e Scratch.....	158

Introduzione

Il tema del coding ha iniziato ad interessarmi maggiormente lo scorso anno grazie al corso Laboratorio di Tecnologie Didattiche tenuto dal Professor Andeas Robert Formiconi. Durante le lezioni ho potuto io stessa sperimentare alcuni programmi di coding tra cui Logo, un software che avevo utilizzato nella scuola primaria e del quale ho un ricordo molto positivo. Probabilmente incoraggiata da questa prima esperienza di coding negli anni della scuola secondaria di primo grado, ho svolto un laboratorio di robotica nel quale ho imparato a programmare e a costruire robot utilizzando il kit LEGO Mindstorms NXT. Negli anni successivi mi sono allontanata da questi percorsi finché, rivedendo la tartaruga, non ho riscoperto la mia passione per la programmazione. Ho deciso quindi di provare anche altri software, come Scratch e Code.org, e durante il loro utilizzo mi sono chiesta se questi programmi potessero anche essere utilizzati in classe e quali potenzialità potessero avere per l'apprendimento. Tramite diversi articoli ho iniziato a scoprire come questi programmi non solo aiutano nello sviluppo del pensiero computazionale, ma possono essere utilizzati come sostegno anche per lo sviluppo di altre competenze, in particolar modo, ho trovato interessante l'articolo di Clements e Battista (1990) sull'utilizzo del coding per aiutare i bambini a diminuire le misconcezioni sugli angoli.

Dopo la lettura di quest'ultimo articolo mi sembrava interessante poter verificare le ipotesi dei due autori nella classe dove avrei svolto il tirocinio, questa classe è una classe 2.0 e potevo avere a disposizione gli strumenti necessari per poter utilizzare i due software. Sostenuta dal professor Formiconi e dalla mia tutor scolastica, la quale è stata disponibile e favorevole al progetto, ho deciso di preparare delle lezioni che mi permettessero di confrontare Scratch e Logo, ma soprattutto di verificare se il loro utilizzo può aiutare i bambini a superare, almeno in parte, le misconcezioni che hanno sugli angoli.

Nel primo capitolo, *Coding e pensiero computazionale in ottica educativa*, vengono delineate le definizioni di coding e di pensiero computazionale, evidenziandone lo stretto legame. L'attenzione è posta prevalentemente sul loro

utilizzo nell'ambito educativo, partendo da un'analisi di come gli alunni utilizzano le tecnologie e domandandosi se è possibile parlare di "nativi digitali". Il primo capitolo si incentra, poi, sui presupposti pedagogico-storici, posti alla base delle teorie che oggi sostengono lo sviluppo del pensiero computazionale. In particolare, viene analizzato il pensiero di Seymour Papert, padre del costruzionismo e del linguaggio di programmazione Logo, utilizzato nella sperimentazione. Per capire meglio il suo pensiero sono delineati anche quelli di alcuni autori che lo hanno influenzato maggiormente: Piaget, del quale ha ripreso alcune idee alla base del costruttivismo; Dewey, per le sue teorie attiviste e Minsky, collaboratore di Papert al M.I.T. Il capitolo si incentra poi sulle motivazioni per le quali il coding dovrebbe diventare una competenza base da sviluppare durante il percorso scolastico; viene analizzato il panorama delle esperienze e delle iniziative messe in atto principalmente negli ultimi anni, prima a livello internazionale e poi, incentrandosi sulla realtà Italiana. Vengono descritte le iniziative a livello legislativo, infatti, negli ultimi anni il pensiero computazionale è entrato anche nei documenti ufficiali e il suo sviluppo è diventato uno degli obiettivi da raggiungere per la formazione completa dei cittadini. Infine, vengono proposti alcuni strumenti che sono già stati utilizzati in diverse realtà scolastiche e aiutano nello sviluppo delle competenze digitali: corsi online, linguaggi di programmazione, Robot didattici e iniziative di *coding unplugged*.

Nel secondo capitolo, *I linguaggi di programmazione*, vengono analizzati e definiti i linguaggi di programmazione. Per prima cosa vengono descritte le caratteristiche che accomunano le diverse tipologie di linguaggio e i principali costrutti che le caratterizzano. Il capitolo si focalizza poi sulle caratteristiche che dei due linguaggi scelti per il progetto didattico: Logo e Scratch. Viene analizzata la tipologia di apprendimento favorita per l'utilizzo di questi linguaggi, definita da Papert "bricolage" uno stile di apprendimento basato su prove ed errori che, quindi, porta ad una nuova visione anche dell'errore negli alunni e negli insegnanti; vengono anche descritti gli ambienti di apprendimento che nascono con questi linguaggi: i micromondi. Sono poi analizzate nello specifico le due tipologie di linguaggio scelte, evidenziandone le differenze: Logo è un linguaggio

testuale mentre Scratch è un linguaggio visuale. In seguito sono presentati alcuni studi che mostrano le differenze tra le due tipologie di linguaggio, evidenziandone svantaggi e punti di forza. Infine, si cerca di mostrare la valenza educativa dei due linguaggi per quanto riguarda l'apprendimento della geometria; l'attenzione è posta su alcuni concetti geometrici, in particolare l'angolo. Anche quest'analisi è sostenuta da alcuni studi svolti principalmente sul linguaggio Logo.

Nel terzo capitolo, *Il progetto svolto in classe: coding con Logo e Scratch*, viene descritto e analizzato il progetto che ho ideato in collaborazione con la mia collega Martina Fini. L'obiettivo che ci siamo poste era quello di confrontare due tipologie di linguaggi di programmazione, quella testuale e quella visuale, utilizzando Logo, nella versione LibreLogo e Scratch. Oltre ad osservare le differenze nell'apprendimento del coding, principalmente, il nostro obiettivo era quello di indagare se tramite i due linguaggi fosse possibile migliorare le proprie competenze geometriche e se uno dei due linguaggi le favorisse maggiormente. Abbiamo deciso di incentrarsi sul concetto di angolo, durante lo studio di questo concetto in molti bambini si creano delle misconcezioni che accompagnano il loro intero percorso scolastico. Abbiamo svolto questo progetto nelle classi dove abbiamo effettuato il tirocinio: io in una classe quinta di Lucca e Martina in una classe quarta di Massa. Per poter analizzare le differenze tra i due linguaggi ogni classe è stata divisa in due gruppi: un gruppo ha utilizzato prima il linguaggio Logo e l'altro ha invece iniziato utilizzando Scratch; a metà percorso i gruppi hanno scambiato linguaggio di programmazione. Per raccogliere i dati, relativi alle conoscenze geometriche, abbiamo strutturato tre test che si incentrano principalmente sul concetto di angolo, da somministrare prima dell'inizio delle attività, a metà del percorso e alla fine delle lezioni. Infine, abbiamo anche preparato un questionario, sempre somministrato a fine percorso, per poter raccogliere dati anche su come i bambini hanno vissuto l'esperienza. Il capitolo descrive le attività svolte e poi si incentra sulla presentazione e l'analisi dei dati raccolti sia nei tre test che nel questionario.

Capitolo 1

Coding e pensiero computazionale in ottica educativa

1.1 Coding e pensiero computazionale: una definizione

Il termine “coding” si riferisce alla scrittura di specifici codici che permettono a delle macchine di compiere determinate azioni, molte attività che ormai fanno parte della nostra quotidianità sono regolate da codici: inviare una mail, utilizzare i social network, mandare un messaggio e molto altro.

Nella realtà scolastica il coding sta iniziando a diventare una pratica utilizzata che viene incoraggiata anche nelle *Indicazioni Nazionali per il curricolo della scuola dell'infanzia e del primo ciclo di istruzione del 2012* che dicono:

Quando possibile, gli alunni potranno essere introdotti ad alcuni linguaggi di programmazione particolarmente semplici e versatili che si prestano a sviluppare il gusto per l'ideazione e la realizzazione di progetti (siti web interattivi, esercizi, giochi, programmi di utilità) e per la comprensione del rapporto che c'è tra codice sorgente e risultato visibile. (Ministero dell'Istruzione, dell'Università e della Ricerca, 2012, p. 66)

Per comprendere a pieno il coding e la sua valenza nell'ambito scolastico è necessario evidenziare lo stretto legame tra coding e pensiero computazionale, infatti, l'obiettivo educativo non può essere la mera scrittura di codici per fare funzionare un programma ma deve essere quello di aiutare gli alunni a sviluppare un pensiero computazionale e, di conseguenza, tutte le competenze legate ad esso. Il pensiero computazionale trova infatti nel coding un campo di applicazione ideale, anche se non l'unico, e porta allo sviluppo di *skills* che non sono fondamentali solo per i programmatori e gli informatici, ma sono utili a tutti come evidenzia Jeannette Wing nel suo articolo del 2006 “Computational Thinking”, un documento importante poiché ha portato ad una ulteriore discussione sulla tematica. L'ultimo decennio è stato infatti scenario di un dibattito che ha interessato sia l'ambito della ricerca che quello istituzionale anche se, il termine

“*Computational Thinking*” era già stato utilizzato a partire dagli anni ottanta da Seymour Papert, matematico e informatico sudafricano, nel suo libro “*Mindstorms*” utilizzandolo anche con una connotazione educativa. Solamente con la Wing si ha però una riflessione approfondita, infatti, lei sottolinea che il pensiero computazionale debba essere inserito tra le abilità di base insieme a leggere, scrivere e far di conto (Wing, 2006), la studiosa elenca anche una serie di caratteristiche proprie del pensiero computazionale evidenziandone la sua valenza educativa:

- Il pensiero computazionale comporta la risoluzione di problemi, la creazione di sistemi e capire il comportamento umano, avvalendosi dei concetti fondamentali dell’informatica.
- Il pensiero computazionale è pensare in modo ricorsivo.
- Il pensiero computazionale è usare l’astrazione e la decomposizione quando si trova di fronte ad una composizione complessa o deve creare un grande e complesso sistema.
- Il pensiero computazionale è pensare in termini di prevenzione, protezione e di recupero, prevedendo i casi peggiori.
- Il pensiero computazionale utilizza un ragionamento euristico per trovare una soluzione. Consiste nel pianificare, imparare e saper gestire delle eventuali incertezze.

Queste caratteristiche permettono di vedere ancora più chiaramente le potenzialità di questo pensiero in diversi ambiti e, come sottolinea la stessa Wing, il pensiero computazionale diventerà realtà quando sarà per tutti e in tutti i luoghi.

Dopo la Wing altri studiosi hanno definito il pensiero computazionale, ma ancora oggi non è stata trovata una definizione univoca, diversi autori però concordano sul fatto che sia un nuovo modo di pensare che porta ad una soluzione dei problemi in maniera differente poiché stimola la creatività e il pensiero meta cognitivo. Michael Lodi ha esaminato diverse pubblicazioni sul pensiero computazionale e ha individuato ulteriori concetti che si ripetono nelle diverse definizioni del termine (Lodi, 2014):

- Collezione e analisi dei dati: il processo di raccolta delle informazioni appropriate, e di analisi per dare loro un senso, trovando pattern comuni e traendo conclusioni dai dati stessi.
- Rappresentazione dei dati: il processo di rappresentazione e organizzazione di dati e risultati, sia visiva (grafici, testo o immagini) sia astratta (strutture dati).
- Decomposizione dei problemi: il processo di divisione del problema in parti più piccole e affrontabili.
- Astrazione: il processo di riduzione della complessità, per far emergere l'idea principale mantenendo solo alcuni aspetti e tralasciandone altri.
- Generalizzazione e riconoscimento di pattern: l'abilità di riconoscere come alcune parti di soluzione possono essere riusate nella stessa o riapplicate a problemi simili.
- Algoritmi: una serie ordinata di passi per risolvere un problema o raggiungere un obiettivo.
- Automazione: lasciare ad una macchina i compiti ripetitivi o noiosi, formalizzandoli e facendoglieli eseguire.
- Simulazione, test e *debug*: modellare un processo ed eseguire esperimenti su di esso. Individuare problemi/errori e correggerli.
- Parallelizzazione: organizzare risorse per far loro eseguire task simultanei allo scopo di raggiungere un obiettivo comune.
- Complessità e calcolabilità: individuare un metodo che raggiunga un risultato, possibilmente il migliore e usando meno risorse (tempo, memoria, potenza di calcolo, energia).

Nonostante le numerose definizioni sarebbe necessario trovarne una sistematica nell'ambito educativo affinché si possano stabilire in modo chiaro degli obiettivi educativi e come valutarli. Devono essere infatti ben preparate delle strategie didattiche per aiutare i bambini nell'approccio al coding e, di conseguenza, al pensiero computazionale, per poter scrivere un codice che porti al risultato atteso è necessaria una buona conoscenza del linguaggio utilizzato poiché sono linguaggi definiti da regole precise e per utilizzarli al meglio devono essere

compresi a fondo; spesso si tende a ritenere che i bambini di oggi siano già competenti per tutto quello che riguarda le tecnologie e che sia sufficiente fornirgli un computer e dei programmi e loro apprenderanno autonomamente, ma questo non è sempre vero anzi, spesso la loro conoscenza rimane superficiale.

1.2 Nativi digitali

Il termine “nativi digitali” è diventato ormai di uso comune, enfatizzato dai media che ne parlano spesso in maniera errata rafforzando gli stereotipi e sottolineando come la scuola dovrebbe adeguarsi alle nuove esigenze. L’autore che ha proposto questa definizione è Marc Prensky in diversi scritti pubblicati nel 2001, successivamente sono state proposte diverse definizioni da altri autori: *Net Generation* (Oblinger & Oblinger, 2005; Tapscott, 2009), *Game Generation* (Carstens & Beck, 2005) o *Millenials* (Howe & Strauss, 2000). Questi autori ritengono che le nuove generazioni, immerse nelle tecnologie fin da piccole, siano dei “*native speaker*” dei linguaggi digitali; per quanto riguarda l’età di queste generazioni i numeri cambiano a seconda dell’autore. Secondo Tapscott la data è precisa, gennaio 1977 mentre per Prensky ne fanno parte i nati nella seconda metà degli anni Ottanta, altri autori arrivano ad indicare come data gli anni novanta poiché anche le famiglie sono state raggiunte dalla rivoluzione digitale (Ferri, 2008). Ciò che accomuna i diversi autori è la convinzione che questi ragazzi circondati dalle tecnologie non solo le utilizzino competentemente ma esse influenzino i loro stili cognitivi e di apprendimento. Tra le competenze dei nativi digitali molto discussa è la capacità di *multitasking* sostenuta da diversi autori (Venn & Vrakking, 2006; Tapscott, 2009) e definita da Ferri una “*forma di intensificazione e condensazione nel tempo di esperienze cognitive multiple in forma lineare*”, quindi sono capaci di raccogliere più informazioni contemporaneamente, svolgere diverse azioni nello stesso momento e anche porre attenzione a più fonti o più azioni che li circondano. Il rischio è quello di esporre questi ragazzi ad un sovraccarico cognitivo ma chi sostiene questa teoria non ritiene che sia così poiché, tramite gli strumenti digitali, i ragazzi devono in realtà memorizzare e concentrarsi solo su poche informazioni perché il resto viene immediatamente salvato tramite dei dispositivi. Prensky fa un’ulteriore riflessione

su come la distanza dei nativi dai metodi scolastici è aumentata dal fatto che gli insegnanti fanno parte di quelli che lui definisce “immigrati digitali”, nati prima degli anni Ottanta che si sono dovuti adattare al mondo digitale, e quindi parlano un linguaggio diverso ma soprattutto hanno stili cognitivi differenti, spesso sono diffidenti verso le tecnologie e quindi anche poco competenti nel loro uso.

Nonostante diversi autori sostengano le tesi di Prensky, tra cui in Italia Paolo Ferri (2011) che sostiene che “*La nostra opinione, in proposito, è piuttosto radicale: i nativi digitali esistono e la loro differenza specifica è l’intelligenza digitale*”; altri invece non hanno trovato evidenze su quanto sostenuto da Prensky, una sintesi di questi pensieri è stata fatta da Antonio Fini (2017) che ha ripreso i punti essenziali dall’articolo “The ‘digital natives’ debate: A critical review of the evidence” del 2008 di Bennett, Maton e Kervin evidenziando che:

- Nonostante le nuove generazioni siano immerse nelle tecnologie il loro uso rimane ancora superficiale, limitandosi ai giochi, i messaggi e l’uso dei social.
- La differenza nelle competenze nell’uso delle tecnologie non è generazionale, infatti troviamo diversi livelli di competenza anche all’interno delle stesse fasce d’età, ma è legata a diversi fattori che influenzano il livello di accesso alle tecnologie e il loro uso e le competenze che si possono sviluppare che sono legati al contesto socio-economico, i fattori culturali come dice Fini il “*capitale sociale delle persone*” (*Digital divide*).
- Il “*moral panic*” si riferisce alla diffusa presenza di stereotipi e allarmi sociali che sottolineano come i sistemi educativi e gli insegnanti non riescano a far fronte alle esigenze dei nativi digitali.

Nel 2009 lo stesso Prensky rivede le sue idee sul termine nativi digitali che modifica sostituendolo con “saggezza digitale” che supera l’idea della differenza tra generazioni dando al concetto il valore di obiettivo educativo da raggiungere in un’era digitale nella quale la semplice immersione nelle tecnologie non è sufficiente per diventare competenti. Il saggio digitale è secondo Prensky colui che, indifferentemente rispetto alla sua generazione, utilizza criticamente e in

maniera responsabile le tecnologie digitali.

Nonostante i pareri ancora distanti e spesso opposti su come le tecnologie influenzino l'apprendimento, sicuramente nelle società attuale soprattutto i giovani vivono in ambienti immersi nelle tecnologie ed è necessaria una riflessione su come queste influenzino il loro modo di pensare e di apprendere affinché gli ambienti educativi permettano loro di costruire e sviluppare le competenze necessarie per essere utenti attivi che utilizzano le tecnologie in modo consapevole.

1.3 Presupposti teorici del coding e del pensiero computazionale: Seymour Papert

Uno dei primi studiosi a riflettere sulle potenzialità delle tecnologie per l'apprendimento è stato Seymour Papert, matematico nato in Sudafrica nel 1928 dove ha vissuto fino al 1954 quando si è trasferito Cambridge per continuare gli studi matematici per altri quattro anni, successivamente si è trasferito a Ginevra dove ha incontrato Jean Piaget. Il loro incontro è stato uno dei due più importanti per lo sviluppo del suo pensiero e delle sue riflessioni, si sono conosciuti presso l'università di Ginevra, dove Papert ha lavorato dal 1958 fino al 1963, la loro collaborazione ha portato il matematico ad iniziare ad interessarsi allo sviluppo cognitivo dei bambini.

1.3.1 Piaget e il costruttivismo

Jean Piaget era uno psicologo e pedagogista nato a Neuchâtel nel 1896 e morto a Ginevra nel 1980. Piaget è uno dei fondatori della psicologia genetica, che si interessa dello sviluppo della mente umana, osservato e analizzato partendo dallo sviluppo del fanciullo, infatti, solo analizzando il fanciullo si può comprendere meglio la mente dell'adulto. Negli anni quaranta tiene un corso presso l'università di Ginevra di Storia del pensiero scientifico e in quegli anni si interessa ai problemi di epistemologia. Questi studi lo portano a fondare una nuova disciplina: l'epistemologia genetica. Questa disciplina è lo studio che mette in relazione gli stadi dello sviluppo del pensiero del bambino e l'affermarsi nel corso della storia

di nuove forme di pensiero sempre più evolute.

Secondo la visione dell'autore l'intelligenza ha sia una natura biologica sia una forma di adattamento all'ambiente. L'adattamento avviene in una forma evoluta e complessa, attraverso di esso l'uomo non solo conosce l'ambiente ma interviene attivamente su di esso arrivando anche a modificarlo. L'adattamento non è però un processo stabile ma necessita di continui cambiamenti, il bambino, infatti, tramite l'esperienze che compie nell'ambiente si trova davanti a nuovi bisogni da soddisfare e a nuovi problemi da risolvere. Queste trasformazioni, secondo Piaget, sono legate a due importanti processi: assimilazione e accomodamento.

L'assimilazione è il processo tramite il quale si ha esperienza del mondo esterno attraverso schemi o concetti che sono già in nostro possesso (ad esempio lo schema della suzione) mentre l'accomodamento è quel processo che porta alla modificazione degli schemi, dei concetti, attraverso nuove esperienze che permettono al bambino di ampliare le informazioni che già possiede. Quindi il bambino arriva all'adattamento quando raggiunge un equilibrio tra assimilazione e accomodamento, nel tempo il bambino raggiunge nuove forme di equilibrio che lo aiutano ad adattarsi in modo sempre più efficiente alla realtà, questo porta alla formazione di una diversa organizzazione mentale, nelle varie età dell'infanzia. Piaget individua quattro principali stadi di sviluppo:

1. Stadio senso-motorio (dalla nascita ai due anni circa): questo stadio prende il nome dal fatto che il bambino conosce il mondo attraverso schemi basati sulla percezione (quindi i sensi) e sul movimento (motorio). È una fase di grande sviluppo mentale, le sue azioni diventano intenzionali, inizia a scoprire il rapporto causa-effetto e forma un'idea dello spazio a partire dalle sue azioni. Ma non è ancora in grado di fare progetti o di avere immagini mentali.
2. Stadio preoperatorio (dai due ai sei/sette anni circa): in questo stadio il bambino si avvale dell'immaginazione ma non è in grado di compiere operazioni mentali reversibili. In questa fase il pensiero rappresentativo si affina sempre di più, inoltre, acquisendo il linguaggio verbale, potenzia anche la sua comunicazione e riesce ad apprendere ancora di più nuovi concetti. Tipico di questa fase è il gioco simbolico mediante il quale il

bambino imita gli adulti e, a partire da esso, sviluppa il gioco di costruzione. Un'altra caratteristica tipica di questa fase è l'egocentrismo, il bambino infatti ha difficoltà a valutare oggetti e situazioni dal punto di vista dell'altro. Inoltre, a causa dell'egocentrismo, il bambino tende a non saper distinguere e spesso confondere la propria esperienza interiore con il mondo esterno, portandolo a ritenere che gli oggetti siano esseri viventi dotati di intenzionalità, questo aspetto del pensiero viene definito da Piaget "animismo".

3. Stadio delle operazioni concrete (dai sette ai dodici anni): il bambino riesce a compiere operazioni mentali reversibili quindi, mette in relazione con il pensiero più azioni e ricostruisce all'inverso i processi mentali messi in atto. Grazie a questo tipo di operazioni il bambino può compiere più classificazioni contemporaneamente, variare volontariamente il modo di classificare e diventa in grado di comprendere l'idea di serie. Quest'ultima è fondamentale per capire il concetto di numero e quindi imparare le operazioni aritmetiche di base. Inoltre, in questa fase il bambino supera l'egocentrismo.
4. Stadio delle operazioni formali (dai dodici anni in poi): il pensiero diventa astratto e quindi i ragionamenti si basano sulle ipotesi necessitando solamente dell'enunciazione mentale dei problemi. Grazie a questo tipo di pensiero il bambino può acquisire nozioni più complesse e il pensiero ipotetico deduttivo gli permette un'apertura mentale necessaria per interessarsi a fenomeni sociali.

Le sue teorie hanno esercitato una grande influenza sulla psicologia sia europea che americana e grazie ad esse è stato possibile comprendere meglio il pensiero e le strutture mentali del bambino, necessarie anche per una didattica più efficace.

Gli studi successivi hanno confermato la successione degli stadi individuati da Piaget, anche se la comparsa di certe abilità risulta più precoce rispetto a quanto descritto dallo psicologo.

Inoltre, Piaget viene considerato uno dei padri del costruttivismo poiché teorizza che il bambino costruisce lui stesso e in maniera attiva le strutture della mente quindi, la conoscenza non è indipendente dal soggetto che conosce.

Il costruttivismo si basa su alcuni concetti fondamentali che descrivono una precisa visione della conoscenza (Calvani, 2000; Nigris, Negri & Zuccoli, 2007; Carletti & Varani, 2005):

- La conoscenza non deriva da una semplice assimilazione passiva delle informazioni che vengono fornite al discente in maniera trasmissiva ma è un prodotto che chi apprende costruisce in maniera attiva.
- La conoscenza è sempre *situata*: nel processo di apprendimento il contesto nel quale si svolge è fondamentale e lo influenza.
- Per i costruttivisti la realtà ontologica non è rilevante deve invece essere intesa epistemologicamente (Lodi, 2014). Il soggetto quindi non deve appropriarsi di qualcosa di oggettivo da apprendere ma sia il soggetto che apprende che l'oggetto appreso si definiscono solo nella relazione reciproca (Carletti, 2005). Quindi, come espresso nel primo punto, la conoscenza deriva da una costruzione attiva del significato del quale, tramite il linguaggio, vengono elaborati stimoli, percezioni e sensazioni. Questa costruzione non è solo individuale ma è anche sociale, per questo è importante il contesto, che fornisce alcune chiavi di lettura della realtà e rende comunicabili le conoscenze apprese.

Da questi principi è possibile individuarne altri che il costruttivismo elabora nel campo educativo per promuovere una determinata tipologia di didattica:

- L'apprendimento deve cercare di valorizzare le esperienze dirette degli alunni, invece di incentrarsi su lezioni frontali dove i contenuti vengono trasmessi passivamente rimanendo astratti per chi apprende.
- I significati devono essere costruiti insieme mediante un confronto che porta alla loro negoziazione, valorizzando le relazioni nella classe sia con i docenti che tra gli alunni.
- Nell'ambito della valutazione è fondamentale concentrarsi sui progressi più che sui prodotti. Fondamentale è la visione dell'errore che, si allontana dalla visione negativa tipica del nostro sistema scolastico, acquistando valore poiché permette di osservare e quindi poi analizzare i propri progressi.

1.3.2 Dewey e l'attivismo

La teoria di Piaget ha fortemente influenzato il pensiero di Papert ma non è stata l'unica fondamentale per la formazione delle sue teorie. Un'altra visione, precedente al costruttivismo, che ha caratterizzato il 1900 e della quale Piaget condivide la centralità di chi apprende e dell'ambiente durante il processo di apprendimento, è l'attivismo. Questo nuovo pensiero pedagogico è stato sostenuto principalmente da John Dewey (1859-1952) pedagogista americano la cui teoria ha ancora oggi rilevanza nell'ambito educativo specialmente per quanto riguarda alcune sue convinzioni (Avalle e Maranzana, 2005):

- Nuova idea del ruolo e della funzione dell'insegnante e della scuola, rivolti alla formazione della persona ma con un'ottica sociale poiché, l'educazione degli individui è essenziale per il progresso della società.
- I contenuti didattici e gli stili di insegnamento devono tenere conto delle caratteristiche psichiche e dalle modalità di apprendimento dell'alunno.
- I contenuti sono sempre inseriti in un contesto socio-economico ben definito.
- Il metodo proposto vede sempre l'alunno come un protagonista attivo del suo apprendimento.

Dewey vede la scuola come un luogo non preparatoria alla vita ma essa stessa una realtà sociale dove chi apprende deve svolgere esperienze reali, l'educazione quindi diventa un processo di vita non solo una preparazione per la vita futura. La scuola rappresenta la vita sociale e ogni classe viene vista come una "società in miniatura".

Secondo l'autore quindi spetta alla scuola organizzare esperienze che permettano all'alunno di sperimentare la vita sociale reale anche per quanto riguarda il lavoro ritenuto ciò che rende davvero "attiva" la scuola e dove il soggetto sperimenta esperienze reali durante le quali è protagonista del proprio apprendimento. La scuola quindi tramite il lavoro

Fornisce al ragazzo motivo effettivo; gli porge esperienze di prima mano; lo mette in contatto con le cose reali. Oltre a tutto questo, essa viene liberalizzata dalle radici con la sua trasformazione in “valori storici e sociali” e in equivalenti “scientifici”(Dewey, 1973).

Dewey ritiene che tra teoria e pratica ci sia un continuo scambio quindi anche la pedagogia non può essere un sapere fisso e definito ma è un sistema che si arricchisce e si modifica progressivamente grazie all'esperienza. La visione che Dewey ha dell'insegnante è quindi quella di una figura che non si limita alla trasmissione delle conoscenze che possiede, ma affinché possa aiutare lo sviluppo degli alunni, deve avere una valida preparazione perché secondo l'autore “è il leader intellettuale di un gruppo sociale. Egli è il capo, non in virtù di una carica sociale, ma in ragione della sua ampia e più profonda conoscenza e della sua più matura esperienza” (Dewey, 1949).

1.3.3 Papert e Minsky

Il secondo incontro fondamentale per Papert è stato quello con Marvin Minsky, avvenuto dopo il suo ritorno negli USA nel 1963, entrambi lavoravano al M.I.T. (*Massachusetts Institute of Technology*). I due studiosi fondano il “*Artificial Intelligence Laboratory*” e lavorano, insieme ad altri studiosi, alla creazione dei primi sistemi di intelligenza artificiale. Sempre in questi anni Papert lavora alla creazione di un linguaggio di programmazione che possa essere utilizzato dai bambini “Logo”. Lui riteneva che il modo migliore per far comprendere i principi della programmazione ai bambini fosse farli direttamente programmare, usando i computer in modo ludico e costruttivo i bambini, oltre ad apprendere le basi della programmazione, liberano la loro creatività. Questo suo pensiero si oppone a quello di un altro studioso le cui ipotesi erano condivise in quegli anni, Skinner un comportamentista che riteneva invece i computer macchine che insegnavano, che in qualche modo “programmavano” i bambini e avrebbero addirittura sostituito le insegnanti. Il computer forniva quesiti ed esercitazioni in sequenza che aumentavano gradualmente di difficoltà, le risposte corrette venivano adeguatamente rinforzate e venivano seguiti i ritmi di apprendimento di ogni

alunno. Papert invece si allontana da questa visione proponendo un nuovo approccio.

1.4 La teoria del costruzionismo

Partendo dal costruttivismo di Piaget, anche influenzato dalle teorie pedagogiche attiviste e grazie ai suoi studi con Minsk, Papert elabora la sua teoria dell'apprendimento basata sull'approccio multidisciplinare, il "costruzionismo".

La teoria di Papert si fonda su alcuni principi fondamentali (Harel & Papert, 1991):

- Il protagonismo dello studente che come diceva Papert "programma il computer e non si fa da lui programmare", il computer deve incoraggiare lo sviluppo delle abilità cognitive e meta cognitive dello studente non deve insegnare esplicitamente delle nozioni o far diventare gli studenti dei programmatori.
- L' "inversione epistemologica" secondo la quale è necessario sostituire l' "imparare per usare" con l' "usare per imparare", dando maggiore spazio ai momenti di apprendimento.
- Rivalutare il pensiero operatorio concreto rispetto a quello formale logico-deduttivo, Papert ritiene che siano le esperienze concrete e casuali alla base delle teorizzazioni scientifiche.
- Apprendimento sintonico, un apprendimento significativo per lo studente poiché la sua concretezza e operosità coinvolge lo studente anche a livello corporeo.
- Gli ambienti o "micro mondi" papertiani sono definiti "palestre cognitive" dove si alternano un continuo *problem finding* e *problem solving* poiché si cerca di fare leva sui progetti personali e significativi per gli studenti che vengono modificati in itinere dagli stessi allievi.
- "Epistemologia dell'indeterminazione gestita" che si basa su questo continuo procedere per aggiustamenti che si allontana della logica del vero o falso, del giusto o sbagliato.

- “Apologia dell’errore” una vera e propria pedagogia dell’errore che supera la sua visione negativa e lo valorizza.

Come si enuncia dalla lettura di questi principi Papert, con la sua teoria, ritiene la scuola luogo di costruzione e non semplice trasmissione della conoscenza e il computer dovrebbe essere uno degli strumenti di apprendimento che la scuola può utilizzare dove lo studente è protagonista del suo apprendimento. Lo scopo dell’utilizzo del computer è quello di promuovere un apprendimento attivo come dice lo stesso Papert “*dovrebbe essere il bambino a programmare il computer e non il computer a programmare il bambino.*”[...] “*I bambini devono cambiare il loro status di `consumatori' di informazioni in quello di `produttori' di conoscenza.*”. Il computer infatti diventa un mezzo per conoscere, scoprire e costruire, uno strumento tramite il quale si possono svolgere esperienze significative, poiché può permettere di apprendere spontaneamente e di aumentare la fiducia e la consapevolezza delle proprie capacità (Capponi, 2009).

Quindi, il costruzionismo nasce a partire dagli studi di Piaget e dalla sua teoria costruttivista, soprattutto dall’idea che la sola trasmissione di conoscenze da un individuo ad un altro non sia sufficiente per l’apprendimento di queste, con una visione opposta al modello istruzionista che vede la mente del bambino come un vaso da riempire. Piaget e Papert ritengono invece che chi apprende sia costruttore di strutture di conoscenza, Papert sottolinea come affinché sia più significativa ed efficace, la costruzione di conoscenze deve avvenire in un ambiente dove il soggetto è impegnato nella costruzione di cose concrete e che può condividere. Papert chiama questa forma di costruzione mentale “pensiero concreto” che, come affermato nei principi sopra indicati, non è più solo una delle fasi enunciate da Piaget ma risulta fondamentale nell’apprendimento che quindi diventa, come lo definisce Papert, “sintonico” e si fonda su tre principi: *continuità* con le esperienze e le conoscenze pregresse di chi apprende; *potenza* nella realizzazione di esperienze e progetti significativi e *risonanza* culturale delle conoscenze da apprendere (Papert, 1984).

Secondo il costruzionismo la trasmissione di conoscenza deve essere vista più come un “set da costruzioni” prendendo spunto dai set dei Lego, poiché per questa

teoria i bambini devono scoprire da soli le conoscenze di cui capiscono di aver bisogno a partire dal “set” fornito dall’educatore.

Viene introdotto il concetto di “artefatti cognitivi” che sono questi oggetti o dispositivi facilitatori dell’apprendimento che necessitano di poter essere condivisi, discussi ed esaminati. Questa visione sostiene che spesso se si presentano dei problemi durante l’apprendimento, essi derivano non dalla complessità dell’argomento o dei concetti, ma dai materiali proposti che non sono adeguati perché non avvicinano il concetto il più possibile alla realtà e quindi non facilitano l’apprendimento.

Il costruttivismo quindi tende a valorizzare l’apprendimento basato sul *problem solving* e il *problem finding* poiché utilizzando gli strumenti forniti, tra cui LOGO ad esempio, ed i “micro mondi” (vedi capitolo 2) i problemi da risolvere si creano e si modificano facendo, questo porta ad un tipo di apprendimento in cui si procede analizzando di volta in volta i risultati e facendo continui aggiustamenti.

1.5 Il coding e le competenze chiave di cittadinanza

Il coding allora ha una valenza educativa? Dovrebbe essere insegnato a scuola? Sicuramente negli ultimi anni, come è già stato espresso nei paragrafi precedenti, le tecnologie hanno subito un rapido incremento che ha coinvolto molti ambiti della nostra società compresa la scuola. Questi cambiamenti hanno anche portato alla nascita di nuovi linguaggi e alla necessità di sviluppare nuove competenze, come evidenziato da Maria Ranieri (2010) la *literacy* si sta sempre più ampliando ed è nata una nuova competenza: la competenza digitale (o *digital literacy*). Questa nuova competenza oggi è stata ampiamente discussa e sono state proposte diverse definizioni, la prima è quella di Gilster (1997) che la definisce “*l’abilità di comprendere e utilizzare le informazioni in molteplici formati a partire da un’ampia varietà di fonti quando viene presentata attraverso il computer*” (Gilster, 1997, p.1 trad. Ranieri). Questa competenza risulta quindi fondamentale in una società dove, specialmente tra i giovani, le informazioni vengono apprese tramite il computer e le fonti sono spesso molteplici e non sempre affidabili, il ragazzo deve quindi diventare competente nella loro fruizione. Insegnare le

competenze di base è compito della scuola da sempre, “è parte della sua missione canonica” Calvani (2010), quindi anche questa nuova competenza non può che essere sviluppata nell’ambito scolastico. La competenza digitale è, infatti, anche una delle otto competenze chiave dell’apprendimento¹ che sono promosse a partire dalla Raccomandazione del Parlamento Europeo e del Consiglio del dicembre 2006 (2006/962/CE) che aveva come argomento principale l’apprendimento permanente che in una società in continuo cambiamento viene ritenuto fondamentale. Queste otto competenze vengono ritenute basilari da acquisire e sviluppare per la formazione di un cittadino consapevole che possa inserirsi con successo nel contesto sociale e professionale. La quarta riguarda, come sopra detto, proprio le competenze digitali che vengono definite come il

“saper utilizzare con dimestichezza e spirito critico le tecnologie della società dell’informazione (TSI) per il lavoro, il tempo libero e la comunicazione. Essa è supportata da abilità di base nelle TIC: l’uso del computer per reperire, valutare, conservare, produrre, presentare e scambiare informazioni nonché per comunicare e partecipare a reti collaborative tramite Internet”. (Legge 394/10 gazzetta ufficiale dell’unione europea, 2006)

Da esse sono poi stati delineati dal MIUR (Il Ministero dell’Istruzione, dell’Università e della Ricerca), nel 2007, con il “Regolamento recante norme in materia di adempimento dell’obbligo di istruzione” (decreto 22 agosto 2007, n.139) gli obiettivi che devono essere raggiunti al termine dell’istruzione obbligatoria. Tali obiettivi sono divisi in quattro assi culturali (dei linguaggi, matematico, scientifico-tecnologico e storico-sociale) e otto competenze chiave di cittadinanza riportate per esteso:

¹ Le otto competenze chiave del quadro di riferimento europeo sono: 1. comunicazione nella madrelingua; 2. comunicazione nelle lingue straniere; 3. competenza matematica e competenze di base in scienza e tecnologia; 4. competenza digitale; 5. imparare a imparare; 6. competenze sociali e civiche; 7. spirito di iniziativa e imprenditorialità; 8. consapevolezza ed espressione culturale.

1. Imparare ad imparare: organizzare il proprio apprendimento, individuando, scegliendo ed utilizzando varie fonti e varie modalità di informazione e di formazione (formale, non formale ed informale), anche in funzione dei tempi disponibili, delle proprie strategie del proprio metodo di studio e di lavoro.
2. Progettare: elaborare e realizzare progetti riguardanti lo sviluppo delle proprie attività di studio e di lavoro, utilizzando le conoscenze apprese per stabilire obiettivi significativi e realistici e le relative priorità, valutando i vincoli e le possibilità esistenti, definendo strategie di azione e verificando i risultati raggiunti.
3. Comunicare:
 - comprendere messaggi di genere diverso (quotidiano, letterario, tecnico, scientifico) e di complessità diversa, trasmessi utilizzando linguaggi diversi (verbale, matematico, scientifico, simbolico, ecc.) mediante diversi supporti (cartacei, informatici e multimediali)
 - rappresentare eventi, fenomeni, principi, concetti, norme, procedure, atteggiamenti, stati d'animo, emozioni, ecc. utilizzando linguaggi diversi (verbale, matematico, scientifico, simbolico, ecc.) e diverse conoscenze disciplinari, mediante diversi supporti (cartacei, informatici e multimediali).
4. Collaborare e partecipare: interagire in gruppo, comprendendo i diversi punti di vista, valorizzando le proprie e le altrui capacità, gestendo la conflittualità, contribuendo all'apprendimento comune ed alla realizzazione delle attività collettive, nel riconoscimento dei diritti fondamentali degli altri.
5. Agire in modo autonomo e responsabile: sapersi inserire in modo attivo e consapevole nella vita sociale e far valere al suo interno i propri diritti e bisogni riconoscendo al contempo quelli altrui, le opportunità comuni, i limiti, le regole, le responsabilità.
6. Risolvere problemi: affrontare situazioni problematiche costruendo e verificando ipotesi, individuando le fonti e le risorse adeguate,

raccogliendo e valutando i dati, proponendo soluzioni utilizzando, secondo il tipo di problema, contenuti e metodi delle diverse discipline.

7. Individuare collegamenti e relazioni: individuare e rappresentare, elaborando argomentazioni coerenti, collegamenti e relazioni tra fenomeni, eventi e concetti diversi, anche appartenenti a diversi ambiti disciplinari, e lontani nello spazio e nel tempo, cogliendone la natura sistemica, individuando analogie e differenze, coerenze ed incoerenze, cause ed effetti e la loro natura probabilistica.
8. Acquisire ed interpretare l'informazione: acquisire ed interpretare criticamente l'informazione ricevuta nei diversi ambiti ed attraverso diversi strumenti comunicativi, valutandone l'attendibilità e l'utilità, distinguendo fatti e opinioni.

Dopo l'analisi di queste competenze è necessario domandarsi se il coding può avere un ruolo nel loro sviluppo e se il coding stesso possa essere definito una competenza digitale.

1.6 Il coding è una competenza digitale?

Le competenze digitali sono quindi promosse a livello legislativo nella scuola, ma il coding può essere inserito tra di esse? Sicuramente, insegnare il pensiero computazionale attraverso il coding può portare allo sviluppo di diverse delle competenze chiave sopra riportate. Le prime che possono essere legate al coding sono *progettare* e *risolvere problemi*, infatti, quando dobbiamo scrivere un codice per prima cosa è necessario individuare un obiettivo da raggiungere e progettare come farlo. Durante la scrittura però nascono spesso problemi che necessitano di una soluzione o ci troviamo di fronte a degli errori che non permettono l'esecuzione del codice o di raggiungere l'obiettivo prefissato, ma attraverso la programmazione, è possibile sperimentare diversi metodi e tentare sempre nuove strategie per raggiungere l'obiettivo finale. Per riuscire a risolvere il problema è necessario comprenderlo a fondo e analizzare bene la possibile strategia per risolverlo altrimenti il computer, in caso di scrittura errata, non riuscirà ad eseguire il codice. Anche la competenza chiave del *comunicare* è legata al coding

poiché esso utilizza un linguaggio specifico, sia nel caso in cui sia un linguaggio testuale che a blocchi, che deve essere ben conosciuto sia per poterlo comprendere che per poterlo utilizzare per rappresentare il proprio progetto. Inoltre l'approccio al coding dovrebbe basarsi su un apprendimento per scoperta, spesso collaborativo che stimola sviluppando la capacità di *collaborare e partecipare* e soprattutto porta ad *imparare ad imparare*.

Programmare però ha una valenza educativa non solo perché aiuta nello sviluppo delle competenze, ma aiuta anche nella crescita personale dell'alunno. Infatti, sviluppando il pensiero computazionale tramite il coding oltre ad apprendere nuovi concetti si apprendono anche pratiche (Lodi, 2016) :

- Essere incrementali e iterativi: sviluppare un progetto per raffinamenti successivi.
- Testing e debugging: provare il progetto alla ricerca di errori, e poi risolverli.
- Riuso e remixing: costruire qualcosa di nuovo sulla base di programmi già scritti da altri.
- Scomposizione: scomporre un problema in parti più semplici e facilmente risolubili, per ottenerne poi una soluzione completa.
- Astrazione: ridurre la complessità per far emergere l'idea principale tralasciando dettagli inutili e concentrandosi sugli aspetti importanti per la soluzione.
- Riconoscimento di pattern e generalizzazione: riconoscere che alcune parti della soluzione possono essere riusate o applicate a problemi simili.
- Automazione: lasciare che sia il computer a svolgere compiti ripetitivi e noiosi, dopo averlo "addestrato".
- Simulazione: modellare un processo ed eseguire esperimenti su di esso.
- Attenzione all'efficienza, calcolabilità e complessità: individuare una strategia di soluzioni che porti ad un risultato, possibilmente il migliore usando meno risorse.

Da queste pratiche emerge che durante la scrittura di codici è difficile scrivere perfettamente il programma al primo tentativo e spesso richiede una continua revisione, inoltre, l'errore in alcuni casi può portare alla scoperta di soluzioni alternative o ad una riflessione più profonda sul problema questo può aiutare l'alunno ad allontanarsi dal perfezionismo, che spesso è fonte di stress eccessivo ed insoddisfazione per chi apprende, ma anche una maggiore apertura verso gli imprevisti e le sfide. Imparare a programmare può anche diventare un mezzo di espressione personale attraverso il quale comunicare, socializzare e divertirsi ma anche fare arte e creare lui stesso dei videogiochi, magari anche con intenti educativi. Infine, di grande valenza educativa è il fatto che il pensiero computazionale può aiutare lo sviluppo di prospettive nuove, di una visione diversa del mondo che circonda l'allievo ma anche di se stesso (Lodi, 2016):

- Esprimere se stessi: capire che la computazione può essere un mezzo di espressione.
- Connettersi: riconoscere il potere della collaborazione con gli altri nella creazione di qualcosa.
- Farsi domande: utilizzare la computazione per farsi domande sul funzionamento del mondo che ci circonda.
- Saper gestire la complessità e i problemi difficili: non essere sopraffatti da un problema aperto, ma invece affrontarlo un passo alla volta.
- Tolleranza per l'ambiguità e i problemi aperti: rendersi conto che i problemi hanno infinite soluzioni, che richiedono ogni volta idee nuove, personali e creative.

Attraverso il coding è quindi possibile sviluppare diverse competenze e diventa necessario definire come e quando inserirlo nella scuola.

1.7 Perché fare il coding a scuola?

Almeno in parte, sono state esposte diverse motivazioni per le quali sarebbe importante portare il coding nella scuola ma sarebbe anche importante iniziare a introdurlo fin da piccoli a partire almeno dalla scuola primaria, anche se in realtà sarebbe utile fin dalla scuola dell'infanzia. Come già espresso in precedenza, per prima cosa deve essere ben chiaro che inserire in coding nella scuola non può limitarsi al semplice impiego delle ICT (Information and Communication Technologies) che dovrebbero naturalmente portare allo sviluppo delle competenze sopra descritte, l'obiettivo deve essere quello di sviluppare la competenza digitale ma per sviluppare una particolare forma mentis critica, metacognitiva e riflessiva (Calvani, 2010). Inoltre sarebbe necessario che gli insegnanti svolgessero corsi di formazione o che venissero inserite almeno inizialmente figure di esperti che sostengono gli insegnanti in questi percorsi. Stabilite queste premesse, Calvani fa una riflessione sull'età preferibile per insegnare la competenza digitale, lui sottolinea come in realtà le risposte date sono provvisorie essendo pratiche ancora relativamente recenti e ancora culturalmente determinate. Il professore distingue tre fasi principali (Calvani, 2010):

- Dai 3 ai 9 anni: approccio ludico-esplorativo. Si impiegano occasionalmente nuove tecnologie, in un'ottica principale di *learning with*, finalizzata a valorizzare dimensioni generali della personalità: creatività, autostima, motivazione, piacere espressivo, ecc.
- Dai 9 ai 12 anni: approccio sistematico, nel quale la tecnologia deve diventare oggetto di apprendimento affrontata in modo intenso (prevale dunque un *learning about*) . Si affronta la conoscenza delle diverse tipologie di software, le loro specifiche potenzialità e anche la conoscenza di almeno un linguaggio di programmazione.
- Dai 12 anni in avanti: approccio "disciplinare" (o funzionale). La competenza digitale, nelle sue diverse dimensioni, dovrebbe essere sostanzialmente già acquisita a questo livello di età. Le tecnologie sono adesso impiegate per apprendere meglio specifici contenuti, per risolvere

particolari problemi o per il perfezionamento specialistico di conoscenze tecniche.

Secondo l'autore la fase più opportuna per lo sviluppo della *digital literacy* è la seconda (9-12 anni), fase del pensiero operatorio che anche Papert stesso ritiene fondamentale sia per apprendere in situazione che per compiere riflessioni metacognitive sulle esperienze compiute. Resta però fondamentale la fase precedente perché fornisce le basi e soprattutto può stimolare la motivazione che è fondamentale per l'apprendimento.

1.7.1 Insegnanti e coding

Fondamentale è anche analizzare la figura dell'insegnante che svolge un ruolo centrale per l'apprendimento del pensiero computazionale e per il tipo di approccio anche gli alunni svilupperanno verso le tecnologie. I docenti si trovano a doversi confrontare con nuovi metodi di insegnamento e di apprendimento e ad integrare le ICT come supporto nella didattica. Infatti, una reale integrazione delle tecnologie del pensiero computazionale potrà avvenire solamente quando ci sarà una riconciliazione tra insegnanti e computer (Marcinkiewicz, 1993). Secondo Ertmer (1999) esistono due tipi di barriere: quelle esterne (di primo ordine) e quelle interne (di secondo ordine). Le prime sono legate a tutti gli elementi riguardanti gli ostacoli che dipendono da fattori esterni: l'adeguato accesso alla tecnologia, l'accesso a internet, la formazione legata alle tecnologie; queste barriere possono essere superate con finanziamenti ed iniziative adeguate. Il secondo tipo di barriera invece è legata al singolo insegnante, ai suoi atteggiamenti, ai comportamenti e alle aspettative nei confronti dei dispositivi tecnologici nella didattica ma anche al senso di autoefficacia nell'uso delle tecnologie. Questo secondo gruppo svolge un ruolo altrettanto importante per l'insegnamento delle tecnologie ma spesso non viene tenuto in considerazione nelle iniziative proposte, negli ultimi anni però sono nati diversi progetti che stanno cercando di avvicinare le tecnologie e la programmazione anche a chi ha poche competenze in questi ambiti, tenendo conto delle reali competenze dei

docenti che, dovrebbero essere verificate in precedenza per poter sviluppare corsi adeguati. Inoltre, anche le politiche di diversi stati puntano all'avvicinamento delle scuole al pensiero computazionale.

1.8 Proposte per lo sviluppo del pensiero computazionale nella realtà internazionale

L'importanza dello sviluppo di un pensiero computazionale, supportata dalle ricerche degli ultimi anni (tra cui quelle sopra riportate), ha iniziato ad interessare anche le istituzioni educative di diversi paesi. Spesso si parla solamente di insegnamento dell'informatica (*computer science*) ma iniziano ad esserci diverse eccezioni, è stata infatti inserita la programmazione in diversi curriculum scolastici, ad esempio in Estonia, Grecia ed Inghilterra. Il panorama è piuttosto ampio quindi ho deciso di soffermarmi solamente su alcuni esempi; particolarmente interessante è il *National Curriculum* britannico² le cui linee guida e standard sono alla base del sistema educativo dell'Inghilterra, del Galles e dell'Irlanda del Nord. A partire dal settembre del 2013 è stato inserito un curriculum di *Computing* ben strutturato soprattutto poiché è articolato in verticale su tutti gli anni di istruzione sia primari che secondari. In questo curriculum il pensiero computazionale ha un ruolo centrale, gli obiettivi, infatti, non si limitano a quelli più strettamente legati all'informatica *tout court*, ma sono legati a molte competenze trasversali e alle *life skills* che il pensiero computazionale contribuisce a formare. Questa visione è evidente sin dalle prime righe del documento che dicono “*Un'educazione informatica di alta qualità educa i discenti ad usare il pensiero computazionale e la creatività per capire e cambiare il mondo*” (traduzione mia) ed è anche ricavabile dagli obiettivi sotto riportati che mirano a garantire che tutti gli alunni:

²“National curriculum in England: computing programmes of study” pubblicato in data 11/09/2013 e consultabile all'indirizzo: <https://www.gov.uk/government/publications/national-curriculum-in-england-computing-programmes-of-study/national-curriculum-in-england-computing-programmes-of-study> (consultato il 10/08/2018)

- Possano comprendere e applicare i principi e i concetti fondamentali dell'informatica, tra cui: astrazione, logica, algoritmi e rappresentazione dei dati.
- Possano analizzare problemi in termini computazionali e avere ripetute esperienze pratiche di scrittura di programmi per risolvere tali problemi.
- Possano valutare e applicare la tecnologia dell'informazione, incluse tecnologie nuove o sconosciute, analiticamente per risolvere i problemi.
- Siano utenti responsabili, competenti, fiduciosi e creativi delle tecnologie dell'informazione e della comunicazione.

Sempre nello stesso anno, l'8 dicembre 2013, durante la “Settimana di educazione all'informatica” il Presidente degli USA Barack Obama ha lanciato un appello ai giovani, tramite un breve video, per avvicinarli allo studio dell'informatica. Anche l'anno successivo Obama ha promosso questa iniziativa, ma questa volta lui stesso ha deciso di programmare insieme ad un gruppo di studenti. Queste immagini hanno rapidamente raggiunto tutto il mondo, grazie alla loro diffusione tramite i mass media, mostrando l'importanza attribuita a questo tema dall'amministrazione degli USA, ma soprattutto ha dato grande visibilità al coding e alla possibilità per tutti di farlo. Nel 2016 Obama ha anche lanciato l'iniziativa “Computer Science For All” che ha come obiettivo “mettere tutti gli studenti americani, dall'asilo al liceo, in condizione di imparare l'informatica ed essere così attrezzati con le abilità di pensiero computazionale di cui hanno bisogno”.

In quegli anni, sull'onda di queste iniziative, sono infatti nate diverse proposte che cercano proprio di avvicinare i giovani e le istituzioni educative al coding.

1.8.1 Code.org

Progetto diventato un punto di riferimento per la valorizzazione e la diffusione a livello internazionale del pensiero computazionale. Code.org è stato fondato nel 2013 da Hadi e Ali Partovi, ma anche grazie alla partecipazione di moltissimi

partner³: multinazionali dell'informatica e del web (Amazon, Google, Apple, Microsoft, Twitter e Facebook), associazioni informatiche e dell'istruzione (Association for Computing Machinery, Computer Science Teachers Association e Teach for America), Università, istituti di ricerca e scuole.

Gli intenti dell'associazione, messi in evidenza nella pagina iniziale del sito web, sono quelli di ampliare le possibilità di accesso all'informatica, in particolar modo di offrire questa possibilità alle donne e alle minoranze alle quali sono solitamente negate. Le ricerche dell'associazione mostrano infatti come in America nelle scuole superiori il corso “*Advanced Placement in Computer Science*” ha la maggiore disparità di genere rispetto a tutti i corsi, sulla totalità degli iscritti gli uomini sono infatti il 78% , inoltre solo il 13% degli iscritti sono studenti di colore nonostante questi due gruppi rappresentino il 75% della popolazione americana. Code.org sta cercando di colmare questo divario e le statistiche da loro presentate mostrano che per il secondo anno di fila circa il 48% degli studenti appartengono a delle minoranze sottorappresentate⁴.

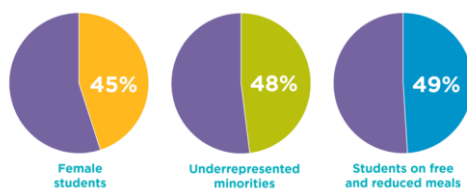


Figura 1 diagramma sugli studenti che utilizzano code.org

Questo però è solo uno degli obiettivi dell'associazione, gli altri riguardano il coinvolgimento del maggior numero di studenti possibile in tutto il mondo (*go global*), ma affinché ciò avvenga è necessario avvicinare le scuole al coding: creando corsi interessanti che vengano promossi dagli insegnanti, ma è anche necessario preparare degli insegnanti di informatica che capiscano il valore di queste pratiche e che le sappiano promuovere agli alunni in maniera competente. Alla base però diventa necessario cambiare i curriculum scolastici e promuovere politiche a favore dello sviluppo dell'informatica e del pensiero computazionale. Questi obiettivi sono in parte stati raggiunti infatti decine di milioni di utenti hanno partecipato agli eventi proposti dall'organizzazione, molti insegnanti hanno

³ <https://code.org/about/partners> (consultato il 12/08/2018)

⁴ <https://code.org/diversity> (consultato il 12/08/2018)

svolto i corsi di preparazione e coinvolto le loro classi e le scuole in questi progetti⁵.

L'offerta didattica proposta da Code.org comprende: "L'Ora del Codice", la sezione "Studenti", risorse e materiali per gli insegnanti e una sezione dove è possibile vedere i progetti che sono stati realizzati dagli utenti.

1.8.2 Hour of code

Una delle iniziative che ha riscosso maggiore successo di Code.Org è "Hour of code", questo progetto nasce con la finalità di avvicinare le persone al coding con una breve lezione introduttiva dalla durata di un'ora. Questa lezione permette di iniziare a capire cosa significa programmare facendolo, questo permette di superare quella distanza che si è creata con chi non ha molte conoscenze informatiche e quindi vede il coding come qualcosa di impossibile e di troppo complicato. Infatti, tutti possono partecipare all'Ora del Codice e in qualsiasi periodo dell'anno anche se il progetto mira principalmente ad avvicinare le scuole alla partecipazione. Nonostante sia sempre possibile partecipare l'evento è in particolar modo promosso durante la "Settimana di educazione informatica" che viene svolta ogni anno durante il periodo della nascita di Grace Murray Hopper, ammiraglio della Marina Militare americana e pioniere dell'informatica, che cade il 9 dicembre. L'ora di codice si presenta come un gioco, che ha valenze didattiche, strutturato in diversi livelli che aumentano di difficoltà ma in modo graduale. Ogni livello presenta un personaggio che deve muoversi in uno specifico scenario per raggiungere un obiettivo, l'utente deve quindi scrivere il codice corretto, utilizzando i blocchi che gli vengono proposti. La schermata si presenta divisa in tre sezioni principali: sulla sinistra troviamo l'area di gioco, dove vediamo l'esecuzione del programma; al centro è situata la *tool box*, dove troviamo i soli blocchi utili per l'attività e a destra è presente l'area di lavoro dove trascinare i blocchi. Quando l'utente completa correttamente il codice passa al livello successivo, al termine dei livelli è possibile salvare un attestato che

⁵ Dati consultabili all'indirizzo <https://code.org/about>.

certifica la partecipazione all'evento. Le attività sono supportate da dei brevi video che forniscono una spiegazione sia su quello che dovranno fare che sul progetto e la sua importanza. Tramite lo svolgimento di questa attività, che può sembrare un gioco, gli utenti imparano le basi dei principali costrutti della programmazione, inoltre, come specificato negli obiettivi, si avvicina alla programmazione e può sviluppare un interesse per questa pratica. Infine, per avvicinare ancora di più i bambini, alla prima versione proposta il *Labirinto classico* sono stati aggiunti nuovi personaggi e ambientazioni tratti dai più recenti film (*Frozen, Oceania, Star Wars*) e videogiochi (*Minecraft, Angry Birds*), che possono utilizzare poiché le aziende detentrici dei relativi diritti commerciali sono tra i partner di Code.org. Queste diverse proposte sono anche suddivise a seconda del livello di programmazione posseduto dall'utente e, essendo pensato per le scuole, è diviso per classi.

1.8.3 La sezione “studenti”

Questa sezione è composta da quattro corsi di livello crescente, consigliati a seconda dell'età, ogni corso è suddiviso in diverse lezioni legate a specifici concetti o attività di programmazione (ad esempio sequenze, cicli o condizionali). Inoltre, una sezione “lezioni tradizionali” presenta una raccolta di materiali per una programmazione *unplugged*, cioè da svolgere senza il computer.

Come per l'Ora di Codice la programmazione avviene in un ambiente visuale con un linguaggio a blocchi. Osservando queste attività da un punto di vista didattico possiamo notare come le lezioni, composte da singole attività ognuna con una consegna precisa, finalizzate non solo alla scrittura corretta del codice ma anche ad una scrittura con il minor numero di “righe di codice” per apprendere dei costrutti informatici, ricordano una forma di esercizio; al termine della lezione troviamo spesso, infatti, delle attività di verifica delle competenze acquisite. Appare quindi un contesto fortemente strutturato in cui il compito è ben definito, ma anche come raggiungere l'obiettivo è fortemente guidato poiché esiste solamente un modo corretto per risolvere il problema, solo al termine di alcune lezioni viene lasciato spazio ad un'attività libera. Anche la presenza dei soli blocchi necessari alla scrittura del codice corretto (troviamo in più solamente

comandi della stessa classe come ad esempio se necessario “gira a sinistra” troviamo “gira a destra”) facilita notevolmente il compito e limita la possibilità di esplorare l’ambiente. Questo tipo di approccio risulta però una proposta valida in relazione agli obiettivi delineati dall’associazione poiché in questo modo è possibile avvicinare una maggiore varietà di utenti poiché diventa un’esperienza che tutti possono svolgere, non richiedendo competenze già acquisite, sia per gli alunni ma anche per un insegnante che vuole svolgere l’esperienza nella sua classe.

1.8.4 CoderDojo

CoderDojo⁶ è un movimento aperto e gratuito che ha organizzato centinaia di club indipendenti in tutto il mondo. Il movimento nasce grazie a James Whelton che all’età di diciotto anni, nel 2011, ha creato un computer club nella sua scuola dopo che aveva suscitato interesse verso il coding tra i suoi coetanei; il primo club nasce quindi a Cork (Irlanda) nel 2011. L’anno successivo un imprenditore e filantropo Bill Liao si interessa al progetto e insieme fondano CoderDojo. Il movimento ha come scopo principale quello di avvicinare al coding bambini e ragazzi poiché i due fondatori ritengono fondamentale, in un mondo moderno come il nostro, conoscere i linguaggi di programmazione e sostengono che sia più facile e si impari meglio se si inizia da piccoli a programmare. CoderDojo attribuisce molta importanza al *peer learning*, si avvale di giovani mentori che più che docenti svolgono un ruolo di facilitatore, ma anche all’apprendimento autonomo, cercando di mostrare ai giovani la possibilità di costruirsi un futuro positivo. Le attività dei club variano a seconda del singolo club poiché, essendo gestite liberamente dai volontari (o *mentor*) sono legate alle loro attitudini e all’età dei partecipanti, inoltre, anche gli ambienti e i linguaggi di programmazione possono variare: oltre a Scratch, scelta di solito principale, per i bambini più grandi vengono usati anche HTML, CSS, PHP, Python e JavaScript.

⁶ Il nome è formato dalla giustapposizione del termine inglese *coder*, ossia programmatore, con il termine giapponese *dojo*, la palestra in cui si praticano le arti marziali.

Vengono anche creati club di robotica ed elettronica educativa che utilizzano ad esempio Arduino.

Essendo un movimento “aperto” che si associa ad iniziative locali aprire un club è piuttosto semplice: basta affiliarsi alla rete e rispettare alcuni principi etici quali la gratuità, la condivisione e software *open source*. Esiste uno specifico statuto da rispettare:

- Ci impegniamo a ispirare e supportare i giovani nell’imparare come creare tecnologia.
- Ci impegniamo a sostenere sempre gli interessi dei giovani partecipanti al nostro Dojo e ad assicurare che le migliori pratiche vengano seguite.
- Ci impegniamo a mantenere lo standard più elevato in termini di tutela dei minori nel nostro territorio.
- Ci impegniamo a non far pagare i ragazzi partecipanti né i loro genitori.
- Ci impegniamo a incoraggiare la partecipazione dei genitori al Dojo.
- Ci impegniamo a condividere le nostre conoscenze in modo libero (nel senso di gratuito e aperto).
- Ci impegniamo a condividere le nostre conoscenze all’interno del nostro Dojo e degli altri Dojo.
- Ci impegniamo a prenderci cura e sostenere il buon nome della comunità globale CoderDojo.
- Ci impegniamo a incoraggiare la collaborazione, il tutoraggio tra pari e il lavoro di squadra tra i partecipanti.
- Ci impegniamo ad accogliere i volontari e i ragazzi a prescindere da genere, razza, orientamento sessuale, credo, religione o abilità.

Pur non essendo un’organizzazione educativa formale, spesso collabora con le scuole e può diventare una risorsa per gli insegnanti interessati al coding e al pensiero computazionale

1.9 La storia del pensiero computazionale nella scuola italiana

Nella scuola italiana l'entrata delle tecnologie digitali si può collocare negli anni ottanta, ma in genere riguardava soltanto gli istituti superiori di indirizzo tecnico. Nel tempo hanno coinvolto sempre più tipologie di istituti e ordini di scuola, ma rimanevano insegnamenti legati all'uso del personal computer senza una vera didattica che li guidava, si utilizzavano solitamente per apprendere software di scrittura, archiviazione e fogli di calcolo, nella scuola primaria invece, in alcune realtà, si diffonde Logo. Il primo Piano Nazionale partì nel 1985 il PNI 1 (Piano Nazionale Informatica). Successivamente si inizia ad avere una diversa visione del computer che diventa uno strumento di apprendimento collaborativo che aiuta nella costruzione di conoscenza, ma il pensiero computazionale e il coding devono aspettare le *Indicazioni nazionali per il curricolo della scuola dell'infanzia e del primo ciclo di istruzione* del 2012 per entrare ufficialmente nella scuola anche se solo il *coding* viene esplicitamente inserito nel documento. Questo documento è entrato in vigore a partire dal decreto ministeriale n.258 del 16 novembre 2012 in sostituzione alle *Indicazioni Nazionali* del 2004 e alle *Indicazioni per il curricolo* del 2007. Nel documento del 2012 troviamo scritto nell'ambito delle tecnologie:

“Quando possibile, gli alunni potranno essere introdotti ad alcuni linguaggi di programmazione particolarmente semplici e versatili che si prestano a sviluppare il gusto per l'ideazione e la realizzazione di progetti (siti web interattivi, esercizi, giochi, programmi di utilità) e per la comprensione del rapporto che c'è tra codice sorgente e risultato visibile”.

Inoltre uno degli obiettivi di apprendimento, ma solo al termine della classe terza della scuola secondaria di primo grado è proprio *“Programmare ambienti informatici e elaborare semplici istruzioni per controllare il comportamento di un robot”.*

Per quanto riguarda il termine “pensiero computazionale” dobbiamo aspettare il 2015 la cosiddetta “Buona Scuola” con la legge del 13 luglio 2015, n.107.

Nell'articolo 1 al comma 7 troviamo infatti tra gli obiettivi formativi: “*Sviluppo delle competenze digitali degli studenti, con particolare attenzione riguardo al pensiero computazionale, all'utilizzo critico e consapevole dei social network e dei media nonché alla produzione e ai legami con il mondo del lavoro*”.

1.9.1 Piano nazionale scuola digitale

L'importanza del *coding* e del pensiero computazionale diventa centrale in uno dei pilastri di questa nuova riforma il *Piano nazionale scuola digitale* (PNSD) sempre dello stesso anno. Quest'ultimo nasce dopo il Piano Nazionale del 2007 che aveva come interesse principale quello di modificare gli ambienti di apprendimento avvicinandolo al mondo digitale promuovendo proprio un'innovazione digitale. Le iniziative principali sono state:

- L'azione LIM in classe: mira alla dotazione delle aule scolastiche di Lavagne Interattive Multimediali, uno strumento didattico che ha la funzione di creare lezioni interattive, multisensoriali e condivise sia all'interno della classe che con l'esterno.
- L'azione Cl@ssi 2.0: simbolo dell'idea di nuovi ambienti di apprendimento innovativi, ha riguardato 416 classi dal 2009 al 2012, l'iniziativa è descritta dallo slogan “non più la classe in laboratorio, ma il laboratorio in classe”. L'aula con i suoi strumenti diventa, infatti, un laboratorio in cui fare esperienza diretta e dove i confini si ampliano oltre le sue quattro mura. La classe infatti diventa un contesto aperto e integrato con l'esterno. Innovativa è poi l'impostazione educativa basata sulla personalizzazione dell'apprendimento e sulla costruzione collettiva della conoscenza tenendo però sempre presenti i tempi di apprendimento individuali.
- L'azione Scuol@ 2.0: avviata nel 2011, ha riguardato 14 istituti scolastici modificandone non solo la programmazione didattica ma anche nuovi modelli di organizzazione delle risorse umane e delle infrastrutture.

Il PNSD del 2015 continuando su questa scia ha l'obiettivo di realizzare in tre

anni un'ulteriore innovazione digitale grazie a quattro passaggi principali:

- **Strumenti:** Fornire alle scuole le condizioni per l'accesso alla società dell'informazione, facendo diventare il "Diritto a Internet" una realtà per abilitare concretamente una didattica digitale.
- **Competenze e contenuti:** Trovare competenze digitali comuni da far sviluppare ad ogni studente e sostenere i docenti affinché, con strategie didattiche adeguate, aiutino lo sviluppo di competenze chiave e propongano format didattici innovativi. L'obiettivo è quindi innovare i curricula scolastici.
- **Formazione del personale:** È necessario migliorare le competenze digitali dei docenti creando standard efficaci e continui nel tempo per la formazione all'innovazione didattica sia iniziale che durante tutto il periodo del servizio.
- **Accompagnare la scuola nella sfida dell'innovazione:** Innovare le forme di accompagnamento e propagare le innovazioni, migliorando anche la collaborazione con enti esterni.

Oltre agli obiettivi ogni passaggio è accompagnato da azioni concrete per raggiungerli e vengono indicate le risorse necessarie per realizzarli.

L'azione maggiormente legata al pensiero computazionale è la numero 17 *"portare il pensiero logico-computazionale a tutta la scuola primaria"* il suo obiettivo principale è quello di fare in modo che tutti gli studenti della scuola primaria abbiano almeno una breve esperienza di pensiero computazionale, nel piano vengono proposte almeno dieci ore ogni anno. Viene ritenuto, infatti, centrale partire fin da giovani sia per anticipare la comprensione di questo tipo di logica sia per prepararli a competenze che sono ormai al centro della nostra società e quindi fondamentali sia nella vita quotidiana che nella carriera. Per far in modo che ciò avvenga viene sostenuta e promossa l'iniziativa "Programma il Futuro", ma non è l'unica possibilità, l'idea è quella di sviluppare nuove iniziative per sviluppare il pensiero computazionale coinvolgendo anche la scuola dell'infanzia. Vengono suggeriti anche corsi *unplugged*, programmazioni di droni e l'utilizzo di stampanti 3D.

Un documento successivo al PNSD viene pubblicato il 22 febbraio 2018 e sono le *Indicazioni Nazionali e Nuovi Scenari* questo nuovo documento non vuole sostituire le Indicazioni precedenti, ma ne fornisce una visione ampliata e nuova incentrata sulla rilettura di alcune tematiche come ad esempio l'integrazione tra le discipline di studio, la scelta di metodi didattici efficaci ed operativi e la predisposizione di ambienti di apprendimento coinvolgenti e partecipati. Nel documento troviamo anche un approfondimento sul pensiero computazionale, che mancava nelle indicazioni precedenti, il quale viene inserito tra gli strumenti culturali per la cittadinanza. Il pensiero computazionale viene definito “*un processo mentale che consente di risolvere problemi di varia natura seguendo metodi e strumenti specifici pianificando una strategia*” (Ministero dell'Istruzione, dell'Università e della Ricerca, 2018). Questa definizione sottolinea come questo tipo di pensiero stimoli ad agire consapevolmente scomponendo i vari problemi e trovando metodi idonei per ogni parte individuata, questo modo di pensare è fondamentale per la programmazione di computer o robot, che necessitano di istruzioni precise, ma non è limitato a queste attività, può essere fondamentale anche per attività senza macchine. Infatti, viene evidenziato come il pensiero computazionale contribuisce anche allo sviluppo di altre competenze (ad esempio matematiche e scientifiche) e al rafforzamento dello spirito di iniziativa e aiuto anche nel migliorare le competenze linguistiche.

1.9.2 PON

Il PON (Programma Operativo Nazionale) chiamato “Per la Scuola – competenze e ambienti per l'apprendimento” è un piano di interventi varato dal MIUR di durata settennale dal 2014 al 2020. Lo scopo è quello di svolgere interventi per l'innovazione organizzativa e didattica nelle scuole per elevare la qualità del sistema di istruzione e formazione, tali interventi sono finanziati dai Fondi Strutturali Europei.

Una delle iniziative più rilevanti è quella del marzo 2016, il MIUR ha indetto un bando per finanziare la realizzazione di “atelier creativi”, presidi anche dall'azione n°7 del PNSD denominata “*Piano Laboratori*”. Gli atelier creativi sono luogo ideale per la didattica laboratoriale in cui si incontrano manualità, artigianato,

creatività e tecnologie utilizzando “*robotica ed elettronica educativa, logica, e pensiero computazionale, artefatti manuali e digitali, serious play e storytelling*” (Ministero dell’Istruzione, dell’Università e della Ricerca, 2015).

1.10 Proposte per lo sviluppo del pensiero computazionale nella realtà italiana

Nel corso degli ultimi anni sono stati proposti diversi strumenti che, se utilizzati in maniera competente da insegnanti che li propongono con un’ottica didattica, possono supportare lo sviluppo del pensiero computazionale ed essere utilizzati anche nel contesto scolastico.

1.10.1 Programma il futuro

“Programma il Futuro” nasce a partire dall’anno scolastico 2014-2015 grazie al CINI (Consorzio Interuniversitario Nazionale per l’Informatica) in collaborazione con il MIUR. Questo progetto ha come obiettivo quello di proporre alle scuole degli strumenti che permettano di sviluppare le competenze di base dell’informatica con semplicità, divertendosi e soprattutto che siano facilmente accessibili in modo da poter essere utilizzati da più scuole possibili; anche la semplicità svolge un ruolo fondamentale poiché questi programmi non richiedono una grande conoscenza sull’utilizzo del computer o sulla programmazione e questo permette a tutte le insegnanti e gli studenti di provare queste attività. Questo progetto è stato anche riconosciuto come iniziativa di eccellenza europea per l’educazione digitale nell’ambito degli “*European Digital Skills Awards*” 2016. Vengono utilizzati i materiali didattici prodotti da Code.org che sono stati tradotti in italiano, sono state create due differenti piattaforme web che però sono collegate tra loro: un sito che fornisce spiegazioni, informazioni, approfondimento, video tutorial e dove vengono gestiti gli account (<http://www.programmailfuturo.it>) e un sito dove si possono svolgere le attività didattiche, è infatti la piattaforma di Code.org tradotta in italiano (<https://studio.code.org>). Sono presenti due modalità che forniscono due tipologie di attività didattiche, viene definita “modalità base” quella che consiste nelle

single lezioni dell’Ora di codice e “modalità avanzata” che permette di seguire i corsi completi che sono pensati sia per una versione “classica” con il computer che nella versione *unplugged*. Questo progetto è pensato per delle classi, specialmente della scuola primaria, infatti, un insegnante può registrarsi e creare una classe sulla piattaforma, ma è possibile anche seguire le lezioni come singolo. Per quanto riguarda la proposta offerta valgono le osservazioni fatte per Code.org (spiegata nel paragrafo 1.8.1).

1.10.2 CoderDojo in Italia

In Italia il movimento CoderDojo si sta diffondendo ampiamente, sono attivi oltre quaranta dojo su tutto il territorio nazionale dal 2012⁷. In alcune regioni i dojo hanno collaborato con le associazioni presenti sul territorio organizzando anche corsi di formazione per l’insegnamento del pensiero computazionale destinati ai docenti.

1.10.2 CodeMOOC

Quando si parla di MOOC (massive open online course) ci si riferisce a corsi di formazione a distanza ai quali tutti possono partecipare. In Italia Alessandro Bogliolo ha creato CodeMOOC un corso di formazione certificato dalla durata di quarantotto ore che viene erogato gratuitamente tramite la piattaforma europea EMMA⁸ (European Multilingual MOOC Aggregator), *“il corso è ambizioso e semplicissimo, ha l’obiettivo di aiutare gli insegnanti di qualsiasi scuola e di qualsiasi disciplina ad introdurre il pensiero computazionale in classe attraverso il coding, usando solo attività intuitive e divertenti da proporre direttamente agli alunni”* (Bogliolo, 2016). L’obiettivo non riguarda insegnare metodi e concetti che poi l’insegnante utilizzerà in classe, chi partecipa al corso proverà direttamente attività di coding che poi potrà riportare in classe. Queste attività sono pensate per chi non ha molte competenze informatiche, sono intuitive e

⁷ elenco completo consultabile all’indirizzo: <http://www.coderdojoitalia.org/> .

⁸ Era una piattaforma sperimentale oggi confluita in Federica.eu

proposte in maniera ludica e divertente. Le proposte del corso si avvalgono di diverse risorse tra cui Code.Org e Scratch mentre per le attività *unplugged* si appoggia a CodyRoby e CodyWay.

1.10.3 CS First

CS First è un programma pensato per gli studenti tra i nove e i quattordici anni che fa parte delle iniziative promosse da Google per l'insegnamento dell'informatica. All'interno della piattaforma si possono trovare attività dalla durata di un'ora che riprendono l'idea dell' "ora di codice" o dei corsi a tema divisi in otto attività per un totale di circa dieci ore; ogni attività è supportata da dei brevi video. Il linguaggio di programmazione utilizzato è Scratch, anche in questo caso è possibile per un insegnante creare un club gratuitamente e in questo modo riceverà anche un kit che comprende: delle unità di apprendimento, un opuscolo per i docenti, dei passaporti per gli alunni e alcuni gadget.

1.10.4 Logo

Logo è un linguaggio di programmazione testuale ideato da Seymour Papert negli anni 80. Questo programma è il primo pensato appositamente per bambini e ragazzi e realizzato come uno strumento di apprendimento è, secondo il suo inventore, uno strumento per pensare e non più per insegnare, un ambiente orientato alla soluzione di problemi e alla costruzione attiva di conoscenza. La prima versione di LOGO consisteva in un robot, la cui forma ricorda una tartaruga, dotato di ruote e collegato al computer da un semplice cavo. La tartaruga eseguendo i comandi iniziava un movimento e lasciava un tratto che permetteva di disegnare, ma negli anni ottanta, con l'arrivo dei computer nelle case, Logo divenne un vero e proprio software e la tartaruga una semplice raffigurazione sullo schermo del computer. Inizialmente, infatti, era solo un piccolo triangolo al centro dello schermo che si muoveva eseguendo semplici stringhe di programma creando figure anche molto complesse. Oggi la struttura è rimasta la stessa ma ad eseguire le righe di codice abbiamo una figura che

immediatamente associamo ad una piccola tartaruga, costituita da diverse figure geometriche. Una volta appreso il linguaggio della tartaruga l'alunno è libero di disegnare quello che riesce ad immaginare.

1.10.5 Scratch

La prima versione di Scratch nasce nel 2006 grazie a Mitchel Resnick, collaboratore di Seymour Papert, al “*Lifelong Kindergarten Group*” dei Media Lab del MIT di Boston. Il termine Scratch è legato al verbo “*to scratch*” (graffiare) al movimento dei DJ di girare i dischi musicali sotto le testine di lettura dei giradischi quando mixano la musica. Il “mixare” è infatti alla base di Scratch, ma in questo caso si intende il mescolare contenuti multimediali, immagini, animazioni, suoni e testi con lo scopo di produrre contenuti informatici. Con Scratch puoi infatti creare animazioni, storie interattive, quiz, videogiochi, video musicali e qualsiasi cosa può nascere dalla tua fantasia. L'obiettivo però non si limita alla sola creazione dei progetti, come evidenziato dal motto stesso “immagina, programma e condividi”, Scratch mira a creare una comunità di condivisione dei progetti, invitando anche gli utenti a creare dei remix di quelli già esistenti. I progetti creati hanno un codice aperto e quindi visibile agli utenti selezionando l'opzione “guarda dentro”, inoltre, la creazione di una comunità e lo sviluppo di collaborazione tra utenti vengono sostenuti anche dalla possibilità di commentare i progetti condivisi; al momento sono stati condivisi più di trentotto milioni di progetti⁹. Scratch è pensato specificatamente per una fascia di età che va dagli otto ai sedici anni, ma può essere utilizzato da tutte le fasce di età, oltre ai singoli utenti l'utilizzo è pensato anche per le scuole. Un insegnante può, infatti, creare una classe su Scratch senza che gli alunni necessitino di una mail personale, i progetti possono essere condivisi ma non è obbligatorio quindi l'insegnante può decidere come gestire la sua classe su Scratch. Il linguaggio di Scratch è a blocchi e quindi per creare i codici è sufficiente trascinare i blocchi, modificandone alcuni parametri, sull'area degli *script* e incastrarli nel modo corretto. Oltre all'applicazione online è possibile anche scaricare una versione che

⁹ dati ricavati dall'indirizzo <https://scratch.mit.edu/> (consultato il 26/08/18)

può essere installata sul computer, anche se hanno alcune differenze.

1.10. 6 Robot educativi

Nella scuola si stanno diffondendo anche altre tipologie di applicazione del coding come i robot educativi. Innanzitutto è necessario definire cos'è un robot, il professor Baldi (2015) lo definisce una macchina che può eseguire diversi tipi di compiti in maniera autonoma, è formato da tre parti che spesso vengono fatte corrispondere a quelle dell'uomo perché svolgono funzioni simili: computer e software (cervello), motori e telaio (muscoli e scheletro) e sensori (organi di senso). Il computer e il software hanno il compito di elaborare i dati forniti dai sensori per poi prendere decisioni che fanno muovere i motori. Il concetto di robotica educativa supera il semplice utilizzo del robot a scuola, è un nuovo tipo di approccio che modifica la tipologia di insegnamento. I robot non vengono utilizzati solamente per insegnare la programmazione applicata ai robot o come funzionano e come costruirli, permettono invece, a seconda delle scelte didattiche sul loro utilizzo, di sviluppare diverse competenze, la creatività e la logica degli alunni. L'approccio rimane pressoché quello del coding, infatti, anche i robot per poter funzionare devono essere programmati e solitamente utilizzano linguaggi visuali spesso simili a quelli più famosi come Scratch e quindi per quanto riguarda lo sviluppo del pensiero computazionale le competenze che possono sviluppare sono praticamente le stesse.

In seguito verranno analizzati alcuni dei robot maggiormente utilizzati in ambito educativo:

- Bee-Bot e Blue-Bot: la Bee-Bot (figura 2¹⁰) è un robot educativo la cui forma si ispira ad un'ape (da cui il nome) è stato pensato come strumento didattico per la scuola dell'infanzia e per i primi anni della scuola primaria. La Bee-Bot è in grado di memorizzare semplici comandi, fino ad un massimo di quaranta, e poi muoversi di conseguenza. I comandi si trovano sul dorso del robot: avanti, indietro, svolta a sinistra e svolta a destra. Ogni passo corrisponde a 15 cm e la rotazione è di 90°, inoltre ci

¹⁰ Immagine consultabile all'indirizzo: <https://www.campustore.it/robotica-educativa-elettronica-coding/bee-bot-blue-bot/bee-bot-ricaricabile.html> (consultato il 20/08/2018)

sono altri tre comandi: “pause” che ferma la Bee-Bot per un secondo; “clear” che cancella la memoria e “go” che avvia il programma. Per aiutare i bambini a capire se il comando è stato appreso vengono emessi suoni e luci. La Blue-Bot (figura 3¹¹) invece è la nuova versione del robot, la prima differenza che si nota è relativa all’aspetto, poiché il guscio è trasparente e questo permette ai bambini di vedere le componenti interne, ma la differenza maggiore riguarda la possibilità di programmarla tramite app questo consente maggiori possibilità di comandi: può ruotare anche di 45°, può ripetere l’algoritmo e può memorizzare fino a 200 comandi consecutivi.



Figura 2 Bee-Bot



Figura 3 Blue-Bot

- Cubetto (figura 4¹²): è un robot di legno a forma di cubo, come suggerisce il nome, che si muove su due ruote. Cubetto è stato pensato per la scuola dell’infanzia e per i primi anni della scuola primaria, è composto da un robot che viene controllato, tramite Bluetooth, e da una console costituita da una tavoletta forata dove possono essere inseriti i comandi. Questi ultimi sono sedici tasselli di quattro colori diversi ai quali corrisponde un preciso movimento del robot: avanti, destra, sinistra e più azioni contemporaneamente. I comandi oltre al colore diverso possono essere riconosciuti per la forma e per la sensazione al tatto.

¹¹ Immagine consultabile all’indirizzo: <https://www.campustore.it/robotica-educativa-elettronica-coding/bee-bot-blue-bot/blue-bot-13.html> (consultato il 20/08/2018)

¹² Immagine consultabile all’indirizzo: <https://www.campustore.it/robotica-educativa-elettronica-coding/cubetto.html> (consultato il 20/08/2018)



Figura 4 Kit di Cubetto

- Dash and Dot (figura 5¹³): Dash è dotato di ruote e sensori che gli permettono di muoversi in tutte le direzioni evitando gli ostacoli, questo robot può rispondere a comandi vocali, ballare, cantare, registrare suoni e sparare palline con una catapulta. Dispone infatti di tre microfoni e quattro tasti direzionali. Dot nonostante le dimensioni minori è in grado di dare ordini a Dash tramite infrarossi, Dot ha un microfono, dei pulsanti programmabili e alcuni suoni predefiniti, inoltre come Dash dispone di un occhio a infrarossi che emette luci diverse. Entrambi sono programmabili tramite cinque app di difficoltà crescente.



Figura 5 Dash e Dot

- Lego Mindstorms EV3 (figura 6¹⁴): è un kit che permette di costruire fino a diciassette robot, i primi cinque più semplici e gli altri consigliati per programmatori più esperti. Il robot è programmabile tramite un software

¹³ Immagine consultabile all'indirizzo: <https://www.campustore.it/robotica-educativa-elettronica-coding/dash-dot.html> (consultato il 20/08/2018)

¹⁴ Immagine consultabile all'indirizzo: <https://www.campustore.it/robotica-educativa-elettronica-coding/lego-education/mindstorms-ev3.html> (consultato il 20/08/2018)

compreso nel kit, ma è grazie al “mattoncino intelligente” che vengono controllati i motori ed i sensori e può connettersi tramite Wi-fi e Bluetooth.



Figura 6 Kit Lego Mindstorms EV3

- Mbot (figura 7¹⁵): è un kit per robot pensato per bambini dagli otto anni in su, tramite il quale è possibile creare un robot che si muove tramite ruote grazie a due motori e che possiede sensori ottici e ad ultrasuoni. Inoltre, può collegarsi a Bluetooth o Wi-fi (a seconda della versione). Per programmarlo si utilizza un software mBlock derivato da Scratch.



Figura 7 Mbot

1.10.7 Coding unplugged

Quando si parla di “coding unplugged” ci si riferisce alle attività che permettono di sviluppare pensiero computazionale senza utilizzare strumenti tecnologici. Si tratta di attività, spesso ludiche, che utilizzano strumenti tradizionali (ad esempio

¹⁵ Immagine consultabile all’indirizzo: <https://www.campustore.it/robotica-educativa-elettronica-coding/mbot-makeblock/mbot.html> (consultato il 20/08/2018)

carta, penna, filo, eccetera) che possono essere svolti in classe o all’aperto e introducono i concetti principali del coding come i numeri binari, le sequenze e gli algoritmi.

Di seguito sono spiegati due esempi di coding unplugged utilizzati nelle scuole:

- CodyRoby (figura 8¹⁶): uno strumento per fare coding unplugged strutturato come un gioco di carte, le carte contengono dei simboli che rappresentano tre movimenti base: vai avanti, gira a sinistra e gira a destra. Il nome evidenzia le due componenti del gioco, “code” è chi scrive il codice e “Roby” è il robot che le esegue, quindi, chi gioca e usa le carte per dare istruzioni, è code, mentre, chi le esegue è Roby che può essere rappresentato da una pedina sulla scacchiera o può anche essere interpretato da un altro bambino. Tutti gli strumenti del kit possono essere scaricate gratuitamente.

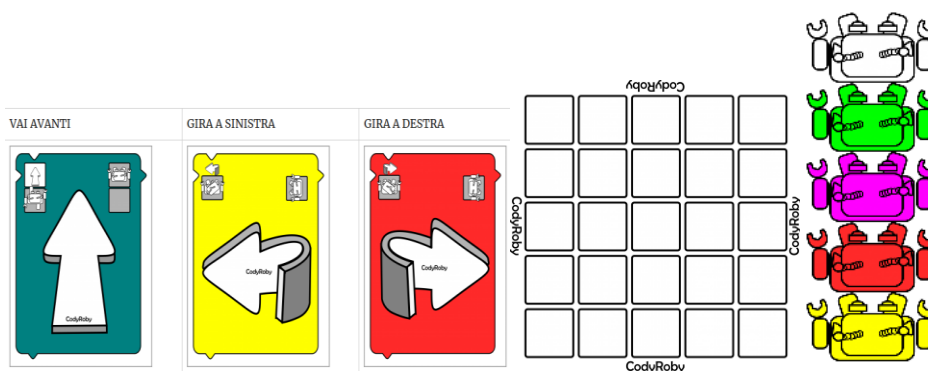


Figura 8 Kit di CodyRoby

- CodyWay (figura 9¹⁷): pensato dal MIUR è un metodo di coding unplugged che si basa su del materiale, al quale tutti possono accedere, da una presentazione condivisa su Google Drive. Il materiale fornito può essere modificato in modo da personalizzare i blocchi e crearne di nuovi. Con questi blocchi è possibile creare percorsi didattici su qualsiasi argomento, si trovano esempi di percorsi pensati per le uscite di sicurezza durante eventi come terremoti o incendi ma anche percorsi per visitare determinati monumenti durante una gita.

¹⁶ Immagine consultabile all’indirizzo: <http://codeweek.it/cody-roby/> (consultato il 21/08/2018)

¹⁷ Immagine consultabile all’indirizzo: <http://codeweek.it/lm05-codyway/> (consultato il 21/08/2018)

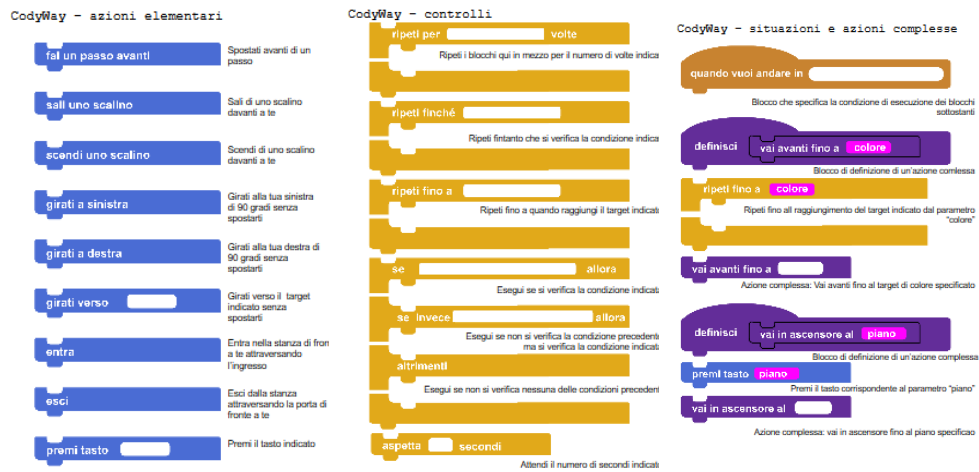


Figura 9 parte del kit di CodyWay

Questi sono solamente due esempi possibili ma le risorse disponibili sono moltissime o è possibile crearle facilmente. Anche programmare il futuro presenta una sezione con diverse attività da fare *coding unplugged*.

Capitolo 2

I linguaggi di programmazione

2.1 Cosa sono i linguaggi di programmazione?

Abbiamo visto cosa significa programmare e il suo valore in ambito educativa, ma come funziona realmente la programmazione? Cosa permette a chi programma è alla macchina di comunicare? Il linguaggio di programmazione svolge proprio questa funzione ed è, quindi, quello strumento che permette di scrivere programmi che realizzano algoritmi. Il termine “linguaggio” nel vocabolario¹⁸ viene definito come la facoltà propria dell'uomo di esprimersi e comunicare tramite un sistema di simboli, in particolare di segni vocali e grafici, può essere un linguaggio verbale o non verbale. Quando parliamo di linguaggio di programmazione questa definizione non basta, è necessario specificare che nella programmazione parliamo di un “linguaggio formale” in cui la forma delle frasi (sintassi) e il loro significato (semantica) sono definiti in modo preciso e algoritmico, utilizza infatti, informazioni chiare e non presenta ambiguità, in altre parole un'informazione non può corrispondere a più significati. Come per il linguaggio naturale¹⁹ però, quello formale è caratterizzato da diverse componenti:

- Dai simboli (alfabeto) e dalle parole (dizionario) che combinati permettono di formare le frasi del linguaggio.
- Dalle regole grammaticali (sintassi) per definire quali sono le frasi corrette composte dalle parole del linguaggio.
- Dal significato (semantica) delle frasi del linguaggio.

¹⁸ <http://www.treccani.it/vocabolario/linguaggio/> (consultato il 02/09/18)

¹⁹ Linguaggio solitamente usato nella comunicazione fra individui di un gruppo sociale che lo condivide; presenta una sua ricchezza espressiva, ma anche sfumature e ambiguità, per cui logica e matematica tendono a ridurne l'utilizzo.

- Per utilizzare correttamente un linguaggio è necessario conoscerne la pragmatica (ad esempio: quali frasi è opportuno usare a seconda del contesto).

Per rendere la scrittura del codice più rapida e semplice sono stati introdotti i “linguaggi simbolici avanzati” che utilizzano le parole e i simboli dei linguaggi naturali che vengono poi tradotti nel linguaggio macchina affinché possano essere eseguiti. Si distinguono due tipologie di linguaggi avanzati: il linguaggio interprete e il linguaggio compilatore. Il primo interpreta singolarmente ogni istruzione impartita alla macchina, ogni volta che traduce una singola istruzione verifica la sua correttezza sintattica, la traduce e infine la esegue ma non rileva se c'è un errore di tipo logico. Di conseguenza l'applicazione è lenta e può crearsi un blocco nel momento in cui si verifica un errore, ma allo stesso tempo questo linguaggio aiuta ad individuare immediatamente quali sono i possibili errori. Il linguaggio compilatore invece esegue tutte le istruzioni insieme e l'operazione di controllo e traduzione avviene una volta sola per tutto il programma, dopo questa operazione si ottiene un codice eseguibile, corretto nella sintassi, ma non sempre nella logica, che può essere direttamente eseguito (Capponi, 2009) .

I linguaggi possono essere quindi diversi tra di loro ma non ha senso cercare di individuare un linguaggio migliore poiché ogni linguaggio viene creato per affrontare determinati problemi e, quindi, può essere più o meno adatto a seconda del tipo di utilizzo che se ne intende fare. Le caratteristiche del linguaggio vengono influenzate anche da altri fattori (ad esempio dal tipo di macchina) ma alcune possono essere attribuite al linguaggio stesso (Ghezzi e Jazayeri, 1994):

Affidabilità (reliability): legata al cercare di minimizzare gli errori di programmazione. Comprende diverse caratteristiche:

- “Scrivibilità”: la possibilità di risolvere un problema nel modo più “naturale” possibile, senza essere distratti da problemi accessori legati a dettagli del linguaggio. Ad esempio è più “scrivibile” un linguaggio ad alto livello rispetto a un linguaggio assembler.²⁰

²⁰ L' Assembler è un linguaggio di programmazione composto di un set "compatto" d'istruzioni di base che permettono di interagire direttamente con i registri interni della CPU; si tratta della forma

- **Leggibilità:** tramite la lettura è possibile seguire la logica del programma e individuare la presenza di eventuali errori. La leggibilità è un criterio soggettivo ma determinati costrutti possono semplificarla notevolmente.
- **Semplicità:** Un linguaggio semplice è facile da padroneggiare e consente agli algoritmi di essere espressi facilmente, ciò rende il programmatore più sicuro del suo lavoro. La semplicità può ovviamente entrare in conflitto con la potenza del linguaggio.
- **Sicurezza:** il linguaggio non dovrebbe fornire funzionalità che rendano possibile scrivere programmi dannosi. Come svantaggio però, le caratteristiche che riducono i pericoli possono anche ridurre la potenza e flessibilità.
- **Robustezza:** è la capacità di gestire eventi indesiderati. Tali eventi possono essere bloccati e si può trovare la risposta adatta che può essere programmata per rispondere al loro verificarsi. In questo modo il comportamento del sistema diventa prevedibile anche in situazioni anomale.

Modificabilità (maintainability): un programma dovrebbe essere facilmente modificabile. Due sono le caratteristiche principali che i linguaggi possono fornire per supportare la modifica e sono strettamente legate tra loro:

- **Fattorizzazione:** la possibilità di modellare una caratteristica che costituisce un'unità logica in un unico segmento di codice, per esempio uso di nomi di costanti, procedure o moduli.
- **Localizzazione:** l'effetto della caratteristica del linguaggio è ristretto in una piccola porzione dell'intero programma.

Efficienza: questo requisito era inizialmente molto importante ma si riferiva principalmente allo spazio e al tempo, oggi si sono aggiunti altri parametri come il costo di produrre e mantenere il software, la portabilità, la riusabilità e così via.

di comunicazione tra uomo e macchina più vicina al linguaggio naturale di quest'ultima.

Dopo aver delineato le basi delle caratteristiche dei linguaggi di programmazione vengono analizzati i due linguaggi che hanno interessato la sperimentazione: Scratch e Logo. Entrambi i linguaggi sono stati pensati per un utilizzo da parte di giovani utenti o comunque utenti non esperti, per questo non richiedono conoscenze preliminari di programmazione. Questi programmi permettono quindi di scrivere procedure elementari ma quello che li caratterizza è anche la possibilità di arrivare a scrivere codici sempre più complessi. Resnick, riprendendo le parole di Papert, sottolinea questo aspetto come uno dei principi per le tecnologie rivolte ai bambini:

Quando si discute di tecnologie per l'apprendimento e l'istruzione, Seymour Papert spesso enfatizzava l'importanza dei "low floors" e dei "high ceilings". Una tecnologia per essere efficace, lui diceva, dovrebbe fornire modi semplici per i principianti (low floors) ma anche modi per loro di lavorare su progetti sempre più sofisticati nel tempo (high ceiling). (Resnick, 2005. Traduzione mia)

Brennan e Resnick (2012), un altro collaboratore del MIT, hanno individuato sette concetti che sono alla base di molti linguaggi di programmazione:

1. **Sequenze:** è un concetto chiave che consiste nella creazione di una serie di singole istruzioni che possono poi essere eseguite. Queste istruzioni e il loro ordine condizionano il comportamento e le azioni che saranno prodotti.
2. **Cicli:** sono un meccanismo che permette di far compiere le stesse sequenze più volte. In questo modo invece di riscrivere le istruzioni più volte possiamo decidere, con un comando solo, quante ripetizioni far eseguire arrivando anche a infinite volte.
3. **Evento:** si verifica solamente quando accade una determinata condizione, se ciò avviene è possibile far derivare un'altra azione da questo evento. Questo lo rende un componente essenziale per un media interattivo.

4. Parallelismo: questa condizione permette a più sequenze di avvenire contemporaneamente e di gestire le possibili complicazioni che ne derivano.
5. Condizionali: è la capacità di prendere decisioni basandosi su determinate condizioni. Questa funzione è fondamentale per permettere al programma di funzionare al meglio.
6. Operatori: forniscono supporto per le espressioni matematiche e logiche rendendo capace il programmatore di fare manipolazioni numeriche come: operazioni matematiche (addizioni, sottrazioni, moltiplicazioni, divisioni e funzioni) ed espressioni.
7. Dati: permettono di immagazzinare, richiamare e aggiornare i valori. Si legano al concetto di liste e variabili.

I due autori hanno poi cercato di selezionare le pratiche che maggiormente aiutano nello sviluppo di questi concetti. Hanno individuato quattro principali tipologie di pratiche: *sperimentare e iterare*, cioè sviluppare un pezzo di codice, provare, svilupparne un altro pezzo, riprovare e continuare in questo modo; *fare test e debug*, controllare quindi che tutto funzioni, individuando e risolvendo eventuali problemi; *riusare e remixare*, queste pratiche permettono di creare qualcosa rielaborando idee o progetti esistenti, infine, *astrarre e modularizzare* che permette di esplorare le connessioni fra il tutto e le sue parti.

Esistono altre caratteristiche che accomunano i due linguaggi, e molti altri linguaggi di programmazione, e che li rendono validi per un progetto educativo.

2.2 Bricolage uno stile di apprendimento adatto alla programmazione.

L'utilizzo di programmi come Logo e Scratch porta allo sviluppo di un apprendimento legato a quello che Papert definisce "bricolage". Secondo Papert l'approccio ai problemi e gli stili di lavoro con il computer sono influenzati da due tipologie di apprendimento diverse, che condizionano il modo di approcciarsi ai

problemi in generale e come chi apprende lavora con il computer: i *planners*, coloro che pianificano un piano di lavoro, e i *bricoulers*, che tendono a procedere per prove ed errori (Papert, 1997). I *bricoulers* definiti da Papert anche *tinkerer*, termine che ha il significato letterale di “riparatore”, hanno quindi un tipo di apprendimento che tende ad avvicinarsi gradualmente alla meta e a procedere per gradi. Il *bricolage* si basa infatti su tre principi: usare le cose di cui si dispone, improvvisare e adattarsi (Lévi-Strauss, 1962). Questa tipologia di *learners* ha uno stile di apprendimento che segue ciò che viene espresso nel cosiddetto *principio di Papert* enunciato da Minsky:

alcuni fra gli stadi più cruciali dello sviluppo mentale sono basati non sulla semplice acquisizione di nuove abilità, bensì sull'acquisizione di nuovi metodi amministrativi per usare ciò che già si conosce[...] poiché una mente non può crescere molto se si limita soltanto ad accumulare conoscenze, ma deve inventarsi anche i modi per poter sfruttare al meglio le conoscenze già possedute (Minsky, 1990).

Il tipo di lavoro che viene svolto durante la scrittura di codici è quindi strettamente vicino al bricolage; tramite questi ambienti non solo l'utente utilizza le istruzioni già presenti ma lui stesso può crearle, adattandosi a nuove situazioni e necessità, e non ci sono differenze tra le due tipologie. Un altro esempio legato al bricolage proposta da Papert (1997) è quello dei Lego poiché quando si costruisce con i mattoncini spesso, si inizia il lavoro senza avere un chiaro obiettivo da raggiungere, ma durante la costruzione si inizia a creare un'idea che diventa sempre più chiara. Questo in parte avviene anche durante l'utilizzo di linguaggi di programmazione, dove le linee di codice vengono continuamente modificate a causa di inconvenienti o di modifiche del progetto che nascono durante il lavoro. Questo permette a chi apprende di sviluppare ragionamenti durante il processo di lavoro invece di limitarsi ad accettarli o scartarli totalmente se li ritiene giusti o sbagliati. Il bricolage poi è legato ad un apprendimento del “far da sé”

Imparare consiste nel riunire una serie di materiali e di attrezzi da poter manovrare e manipolare. Forse, più importante di tutto, è lavorare con ciò che si ha. È un processo che è familiare a tutti noi, a livello conscio, quando, per esempio, affrontiamo un problema empiricamente, tentando di applicarvi tutte le cose che ricordiamo esserci state un giorno di aiuto davanti a problemi simili” (Papert, 1994).

Un altro aspetto che contrappone questa tipologia di apprendimento a quella dei *planners* è la visione dell'errore: chi pianifica vede l'errore come qualcosa di negativo poiché ti allontana dall'obiettivo da raggiungere mentre, chi usa il bricolage utilizza l'errore continuamente valorizzandone gli aspetti positivi.

2.3 L'errore nella programmazione: un momento di apprendimento

Tramite l'utilizzo di programmi di coding, si può, quindi, sviluppare una nuova visione dell'errore. Durante la programmazione il bambino instaura un rapporto diverso con l'errore rispetto a quello che vive quando sbaglia un compito scritto o durante un'interrogazione orale. Durante la scrittura di un codice dopo che è stato fissato l'obiettivo, anche affidatogli dall'insegnante, il bambino ha la possibilità di lavorare in autonomia con il problema partendo dalle conoscenze che già possiede e provando delle idee che secondo lui possono portare al raggiungimento dell'obiettivo, se non funzionano deve riesaminare quanto scritto cercando di capire cosa ha sbagliato, ma non deve necessariamente confrontarsi con una persona esterna che valuta negativamente il suo errore, nasce più un confronto con se stesso viene posta una sfida di *problem solving*. Bisogna sempre tenere conto che il modo in cui l'allievo vive l'errore può essere fortemente influenzato da come chi guida le attività di coding organizza il lavoro e dalla sua stessa visione dell'errore.

Durante la scrittura di codici gli errori possono essere di diverso tipo (Downey, Elkner & Meyers, 2002):

- Errori di sintassi: legati alle regole di composizione del linguaggio, in questo caso il codice non viene eseguito e, solitamente, viene generato un messaggio di errore.
- Errore di esecuzione: si possono vedere solo quando il programma viene eseguito e sono dovuti a degli eventi “non gestibili” che anche in questo caso vengono segnalati.
- Errori di semantica: il programma viene eseguito, ma non fa quello che il programmatore si aspettava. Questi sono gli errori che dal punto di vista dell’apprendimento possono essere i più interessanti, perché possono portare a nuove idee e per essere risolti è necessario riflettere attentamente sui concetti e le procedure computazionali.

Quindi commettere un errore durante la scrittura di un codice non ha solamente conseguenze negative poiché ogni errore è sempre risolvibile, anzi, solitamente porta alla scoperta di soluzioni alternative o comunque permette di capire meglio determinati passaggi del codice, poiché l’alunno deve fare un’ulteriore riflessione su ciò che ha scritto. La revisione del codice porterà alla risoluzione dell’errore, trovando magari strade differenti per raggiungere l’obiettivo prefissato; la soluzione può anche essere trovata confrontandosi con un compagno che potrebbe proporre idee diverse ma sempre valide e questo nuovamente conferma che esistono più soluzioni per un unico problema e non ne esiste solamente una “giusta”.

Secondo Papert infatti l’errore nasce dalla netta distinzione tra giusto e sbagliato che viene vista dall’autore come controproducente poiché i bambini tendono a bloccarsi di fronte all’errore o addirittura dalla prospettiva di poter sbagliare, vedendolo come un insuccesso o una brutta valutazione. “*Il panico dell’errore è la morte del progresso*” diceva Whitehead (Capponi, 2009). L’errore viene quindi rimosso o dimenticato invece di soffermarsi sulla correzione affinché si possa trovare una soluzione, mentre con i programmi di coding la correzione degli errori diviene una parte del processo di comprensione del programma e dei problemi. Far vivere l’errore a chi apprende come qualcosa di negativo riduce l’energia e

l'interesse del bambino che, invece, ha bisogno di non essere rimproverato ma di avere la possibilità di controllare l'errore e di percepirlo come un qualcosa di positivo, in un certo senso amichevole (Montessori, 1966) . Come afferma Papert i bambini

molte volte oppongono resistenza alla ricerca degli errori [...] Un programma è presto scritto e provato. Non funziona. Invece di essere corretto, viene cancellato. Talvolta l'intero progetto è abbandonato. In altri casi, il bambino s'intestardisce, prova, riprova e riprova con ammirevole costanza , ma sempre ripartendo da zero, nell'evidente tentativo di riuscire di un sol colpo. Il bambino può fallire o riuscire a far disegnare la figura all'elaboratore ma non è certo riuscito ad acquisire la strategia di correzione (debugging) (Papert, 1984).

Questo avviene perché, come già evidenziato, spesso la scuola propone una visione degli errori come qualcosa di estremamente negativo e, quindi, il bambino cerca di allontanarsene il più possibile senza cercare invece di soffermarsi a riflettere sui suoi sbagli, quindi spesso anche nel coding trova un'opportunità poter cancellare gli errori senza che ne rimanga traccia. Se invece si acquisisce la modalità di *debugging* si svilupperà un atteggiamento opposto dando una valenza positiva all'errore. Tramite l'esperienza di programmazione è possibile sviluppare questa visione dell'errore, infatti, porta gradualmente a operare le correzioni, invece di eliminare tutto, e a servirsi di sottoprocedure. Altri ritengono che la maggiore tolleranza dell'errore dipenda dal fatto che gli insegnanti che utilizzano Logo siano positivi davanti all'errore e vedano i programmi contenenti *bug* come da modificare non da eliminare. Inoltre come evidenzia Papert stesso “*i bambini imparano che anche l'insegnante è un allievo e che ciascuno di noi apprende dagli errori*” (Papert, 1980). La collaborazione con chi ci circonda e anche con l'insegnante sono importanti ma a condizione che non siano false; con l'utilizzo di LOGO è possibile presentare situazioni inedite anche per l'insegnante che quindi non deve far finta di non sapere. Alunni e insegnanti condividono il problema e

apprendono insieme. Inoltre l'alunno attraverso l'errore è in grado di valutare autonomamente i progressi raggiunti.

2.4 Micromondi: uno spazio dove esplorare i linguaggi di programmazione

Poter esplorare questi linguaggi, compiendo errori e creando cose nuove, è strettamente legato ad un'altra caratteristica di questi linguaggi i cosiddetti "micromondi", ovvero, la possibilità tramite il computer di riprodurre i comportamenti dei sistemi reali. Durante la scrittura di codici è possibile creare piccoli universi, luoghi "sicuri" dove si possono esplorare diverse alternative, sperimentare le proprie ipotesi, inoltre, si possono creare applicazioni, veri e propri percorsi di apprendimento, animazioni, simulazioni geometriche e molto altro.

Il computer diventa quindi un ambiente di apprendimento dove si possono costruire nuove idee e, con il sostegno dell'insegnante, nei micro mondi è possibile imparare tramite la progettazione che permette di esplicitare e discutere teorie sul mondo nel quale gli alunni vivono e con il quale interagiscono. La classe può essere vista, come una piccola comunità nella quale vengono condivise le idee e insieme si può riflettere sull'errore.

Le caratteristiche sopra elencate hanno fornito un'idea di ciò che accomuna diversi linguaggi di programmazione tra cui Logo e Scratch, questi due linguaggi hanno però una sostanziale differenza dal punto di vista operativo: Logo è un linguaggio testuale mentre Scratch è un linguaggio visuale.

2.5 Il linguaggio di Logo

Logo è quindi un tipo di linguaggio testuale, per creare i codici è necessario scrivere ogni singolo comando come se fosse un testo. La versione che ho utilizzato è LibreLogo che unisce il programma Logo al *word processor* Writer che fa parte di LibreOffice, può essere paragonato a Word di Microsoft Office, ma in questo caso si tratta di un software libero (Formiconi, 2018). Essere un

software libero fornisce un'ulteriore valenza educativa a questo programma, sottolineando ancora una volta l'importanza della condivisione e della collaborazione. Per prima cosa un software libero deve distribuire il suo codice sorgente, ma non basta devono anche essere rispettati quattro tipi di libertà:

1. Libertà di eseguire il programma come si desidera, per qualsiasi scopo.
2. Libertà di studiare come funziona il programma e modificarlo in modo da adattarlo alle proprie necessità.
3. Libertà di ridistribuire copie in modo da aiutare il prossimo.
4. Libertà di migliorare il programma e distribuirne pubblicamente i miglioramenti eventualmente apportati, in modo che tutta la comunità ne tragga beneficio.

LibreOffice è quindi stato arricchito da molti utenti che hanno aggiunto numerosi *plugin*²¹ tra cui LibreLogo creato da Németh László che ha inserito le funzionalità di Logo in LibreOffice, incrementandolo poiché ha scritto il *plugin* in linguaggio Python²². Utilizzare Logo diventa molto semplice basta attivare la toolbar e poi scrivere un codice, utilizzando il linguaggio di Logo, nello stesso modo in cui scriviamo il testo di un qualsiasi programma di scrittura e poi eseguirlo con l'apposito tasto, con un codice corretto la tartaruga esegue un disegno che poi potrà essere gestito e manipolato come una qualunque forma grafica di LibreOffice (Formiconi, 2018). Il disegno ottenuto in realtà è composto da tanti oggetti geometrici e viene quindi definito di tipo "vettoriale".

I comandi principali per far muovere la tartaruga sono quattro: "Forward", accompagnato dopo lo spazio da un numero, fa muovere la tartaruga in avanti in linea retta; "Back" fa muovere la tartaruga sempre in linea retta ma invece di andare in avanti torna indietro; "Left", seguito da un numero sempre dopo uno spazio, fa girare la tartaruga a sinistra e infine "Right" la fa ruotare a destra; affinché vengano eseguiti deve poi essere premuto il tasto play sulla *toolbar*.

²¹ componenti aggiuntivi per aggiungere nuove caratteristiche ad un programma, sono estensioni dei file.

²² Python è un linguaggio di programmazione dinamico orientato agli oggetti utilizzabile per molti tipi di sviluppo software.

Altri comandi fondamentali, situati anche sulla *toolbar* sono “home” che riporta la tartaruga al centro della pagina con la testa puntata verso l’alto e “clearscreen” che cancella i disegni fatti. Questo tipo di linguaggio richiede un’attenta scrittura del codice, ha una precisa sintassi con diverse regole da rispettare affinché il codice venga eseguito, basta uno spazio in più o delle virgolette non inserite correttamente per non far muovere la tartaruga, però, viene anche segnalato l’errore e dove si trova rendendo facile e veloce la correzione.

2.6 Il linguaggio di Scratch

Scratch è invece un linguaggio visuale e per poter scrivere il codice è necessario trascinare i comandi già pronti nei blocchi e incastrarli correttamente. Con i linguaggi visuali la sintassi risulta più semplice, ma non è totalmente eliminata, è legata alla forma dei blocchi e agli agganci che possono avere oppure no. Al posto della tartaruga Scratch utilizza lo *sprite*²³ un oggetto grafico sul quale il codice agisce, che può essere il gatto, icona del programma, un altro personaggio messo a disposizione, ma è anche possibile disegnarlo da soli attraverso un semplice programma di disegno a disposizione o importare un’immagine. Lo *sprite* si muove sullo *stage* situato a sinistra della schermata, al centro troviamo invece l’elenco dei blocchi che possono essere utilizzati e a destra l’area script dove si trascinano e impilano i blocchi necessari con la tecnica del *drag and drop*. Il colore e la forma dei blocchi permettono di distinguere immediatamente la tipologia di azioni che possono eseguire, è possibile distinguere diverse forme: gli *hat* (cappello), blocco che viene inserito solo all’inizio del codice; gli *stack* (catasta), blocco che può essere incastrato con gli altri, anche in questo caso la tipologia diversa di incastro aiuta a capire come inserire questi comandi e il *reporter* (corrispondente), blocco che deve essere inserito all’interno di un altro blocco.

I blocchi sono ulteriormente divisi in dieci categorie: *movimento*, riguarda tutto ciò che fa muovere, ruotare o cambiare posizione allo *sprite*; *aspetto*, è legato alle

²³ Termine il cui significato è folletto usato in informatica nell’ambiente grafico per indicare una figura bidimensionale che può essere spostata rispetto allo sfondo (*Stage*)

animazioni che influenzano come appare lo *sprite*; *suono*, permette di eseguire brani o sequenze audio programmabili; *penna*, sono tutti blocchi legati al disegno che viene prodotto dallo *sprite* e che influenzano il tipo di tratto che la penna produce; *variabili e liste*, come dice il nome stesso permettono di creare variabili e liste; *situazioni*, blocchi sulla gestione degli eventi che vengono posizionati all'inizio di una stringa di codice; *controllo*, contiene i blocchi per creare cicli e condizionali; *sensori*, che servono per testare determinate situazioni; *operatori*, necessari per fare operazioni matematiche e logiche e infine *altri blocchi*, per creare nuovi blocchi. Per far eseguire i comandi è possibile o cliccare l'ultimo blocco della linea del codice o possono essere eseguiti in seguito al verificarsi di un determinato evento che può essere ad esempio “*Quando si clicca sulla bandierina verde*”.

2.7 Coding testuale e visuale a confronto

Le due tipologie di linguaggio come descritto mostrano alcune differenze sostanziali, per questo motivo diversi autori hanno cercato di capire se l'utilizzo di un linguaggio testuale o visuale può influenzare l'apprendimento del coding e, in caso affermativo, come lo influenza.

Particolarmente interessante è lo studio di Colleen Lewis (2010), che ha messo a confronto proprio il linguaggio di Logo e quello di Scratch, partendo da una riflessione sul perché Logo non si è diffuso come sperato. L'autrice ritiene che l'ambiente di Logo abbia diverse limitazioni, ad esempio, ogni linea di codice deve sottostare a vincoli sintattici e questo può portare gli studenti a concentrarsi più sulla sintassi del codice che non sul suo significato semantico. Mentre ritiene che utilizzando Scratch si hanno le stesse funzionalità ma non si possono compiere errori sintattici e quindi l'attenzione è focalizzata sulla semantica del codice.

Ipotizza infatti che gli studenti che useranno Scratch:

- Riterranno più facile imparare la programmazione in generale e imparare i singoli costrutti.
- Si sentiranno più sicuri delle loro competenze nella scrittura dei programmi per computer e saranno più propensi a continuare la programmazione con il computer.
- Saranno più abili nel controllo dei cicli e delle condizionali.

Il corso ha avuto una durata di trentasei ore divise in dodici giorni. I partecipanti avevano tra i dieci e i dodici anni. Gli alunni sono stati divisi in due gruppi, che hanno svolto le stesse attività, ma un gruppo ha per i primi sei giorni utilizzato Scratch e l'altro prima Logo. Il lavoro veniva svolto in coppie ed i primi sei giorni dedicati allo studio dei costrutti principali della programmazione.

Le conclusioni sono state diverse da quelle preventivate da Lewis: con Logo si ha un maggiore supporto nello sviluppo della sicurezza e dell'interesse nella programmazione, inoltre i cicli vengono compresi meglio; con Scratch si apprende meglio, in parte, ciò che riguarda le condizionali. Tenendo conto dell'apprendimento in generale del coding i risultati sono quindi molto simili e non viene evidenziata una sostanziale differenza tra le due tipologie di linguaggio. Un'altra ricerca che permette di fare riflessioni interessanti è quella di Weintrop e Wilenky (2015). Questa ricerca ha, però al centro degli studenti della scuola superiore e vuole rispondere a queste tre domande:

- Gli studenti delle scuole secondarie pensano che programmare con un linguaggio a blocchi sia più semplice rispetto a quello testuale e, in caso affermativo, perché?
- Quali sono le differenze che gli studenti delle scuole secondarie percepiscono tra un programma basato su un linguaggio testuale e uno sul linguaggio a blocchi?
- Quali potenziali aspetti negativa hanno identificato gli studenti delle scuole secondarie in un programma basato su un linguaggio a blocchi?

Lo studio ha coinvolto tre classi della scuola secondaria per dieci settimane nel corso dell'anno scolastico; le prime cinque settimane sono state dedicate al lavoro con un programma basato su un linguaggio a blocchi poi, le successive cinque, il lavoro è stato incentrato su un linguaggio testuale; in questo caso i linguaggi utilizzati sono Java e Snap!. I risultati raccolti enunciano che inizialmente i ragazzi hanno trovato più facile un tipo di linguaggio a blocchi poiché risulta più semplice da leggere, anche la forma stessa dei blocchi semplifica il lavoro poiché fornisce degli indizi su come utilizzarli e come incastrarli tra di loro, diviene quindi più facile comporre il testo e permette anche di fare meno errori, infine avere la possibilità di vedere tutti i blocchi diventa anche una fonte di idee e ti mostra quello che potresti fare. Successivamente gli studenti hanno però anche individuato alcuni svantaggi di questo tipo di linguaggio: è meno potente, in questo caso la potenza è legata alle possibilità di fare più cose diverse e a livelli più alti di difficoltà *“i blocchi sono limitanti, non puoi fare tutto quello che fai con Java, immagino. Non esiste un blocco per tutto.”*; programmazione più lenta e codice più lungo, spesso per poter scrivere una funzione diversa con i blocchi ne devi combinare moltissimi e diventa difficoltoso gestire il codice *“se vuoi un blocco specifico e non è presente, dovrai inserire molti blocchi insieme per crearlo e fargli fare quello che vuoi, e penso che con JavaScript è solo una frase”*; infine, non è autentico, cioè viene percepito come qualcosa di utile solo nell'ambito educativo ma non valido in altri ambienti *“Java è un vero codice, mentre, Snap! è qualcosa con cui nessuno ti farà scrivere codici”*.

2.8 I linguaggi di programmazione come supporto nell'apprendimento dei concetti geometrici

Diversi autori hanno indagato su come i linguaggi di programmazione potessero avere una valenza nell'aiuto dello sviluppo di competenze matematiche e geometriche, il nostro studio si incentra sulla geometria e in particolare sul concetto di angolo.

Per quanto riguarda Logo, Papert stesso parla di una geometria della tartaruga, è un nuovo stile di fare geometria, uno stile *computazionale* (Papert, 1984). Il

concetto principale su cui si basa questa geometria è la tartaruga stessa che Papert paragona, per valore, al punto nella geometria euclidea. La tartaruga, al contrario del punto, viene compresa più facilmente da chi apprende poiché può creare un riferimento con qualcosa di conosciuto, non è priva di proprietà e non è statica ma dinamica. Infatti, la tartaruga oltre ad avere una posizione ha anche una direzione definita che le permette di essere più facilmente paragonata a un essere vivente conosciuto dal discente, lui stesso potrebbe fingere di essere la tartaruga ed eseguire i suoi stessi movimenti. Questo nuovo stile geometrico si fonda sui principi del costruttivismo, non cerca quindi nuovi modi di insegnare in maniera trasmissiva la matematica e la geometria ma aiuta il bambino a ricostruirle e a creare lui stesso le conoscenze; infatti questo tipo di metodo oltre alle conoscenze matematiche sviluppa quella che Papert definisce conoscenza matematica, cioè la conoscenza dell'apprendimento stesso.

Papert evidenzia tre principi base della geometria della tartaruga (Papert, 1984):

- Principio di continuità: il primo passo nella comprensione di un concetto nuovo deve essere la sua integrazione con le conoscenze pregresse di chi apprende.
- Principio di potenza: l'ambiente di apprendimento deve stimolare chi apprende affinché possa elaborare progetti carichi di significato ai quali non avrebbe mai pensato prima.
- Principio di risonanza culturale: ciò che apprendiamo deve avere un senso anche all'interno dell'ambiente sociale nel quale ci troviamo.

Questo tipo di geometria non deve essere solamente confinata nell'ambiente di Logo ad esempio anche Scratch se utilizzato per far muovere e disegnare lo *sprite* può presentare le stesse funzionalità nell'ambito geometrico. Durante la creazione di figure geometriche il bambino deve riflettere attentamente sulle caratteristiche che possiedono per poterle riprodurre in modo corretto, particolare attenzione deve essere rivolta alla misurazione dell'angolo, per questo motivo l'utilizzo di questi programmi dovrebbe portare ad un miglioramento sul concetto di angolo.

Diversi studi hanno legato la geometria a questa tipologia di coding e vorrei analizzarne alcuni dei più interessanti. Il primo studio è quello di Richard Noss

(1987), secondo l'autore questo tipo di geometria permette ai bambini di lavorare partendo dalle proprie concezioni ed intuizioni, inoltre, fornisce un contesto dove i concetti appaiono naturali e significativi per loro. I concetti che l'autore prende maggiormente in considerazione sono quelli della lunghezza dei lati e dell'angolo legati ai quattro comandi principali (FORWARD, BACK, RIGHT e LEFT).

Lo studio ha coinvolto otto classi delle quali, quattro, composte da ottantaquattro bambini, hanno utilizzato Logo per circa un anno e le rimanenti, composte da novantadue bambini, sono state scelte come classi di controllo e quindi non hanno utilizzato Logo. Nel test sono state valutate le conoscenze relative alla lunghezza e agli angoli; per quanto riguarda la lunghezza sono state analizzate tre aree: la conservazione della lunghezza, le combinazioni tra le lunghezze e la misura delle lunghezze. Rispetto agli angoli sono sempre tre gli aspetti analizzati: la conservazione dell'angolo in generale, più specificamente la conservazione dell'angolo retto e la misurazione dell'angolo. I risultati dello studio hanno mostrato, relativamente alla lunghezza, che l'utilizzo di Logo ha una tendenza positiva sul concetto di conservazione e della misurazione di questa. I dati relativi agli angoli mostravano risultati maggiori rispetto ai lati ma sempre in due ambiti: la misurazione e la conservazione dell'angolo; la conservazione dell'angolo retto, invece, mostra una tendenza al miglioramento ma senza raggiungere valori significativi.

Un altro studio che ha mostrato risultati significativi è quello di Battista e Clements (1990) i due autori ritengono che l'utilizzo di Logo possa fare da ponte tra l'esperienze concrete e la matematica astratta poiché per disegnare un rettangolo, ad esempio, i bambini devono prima analizzare gli aspetti visivi del rettangolo e riflettere su come mettere insieme le sue parti, ciò richiede un pensiero descrittivo/analitico (Battista & Clements, 1989). L'obiettivo di questo studio era di esaminare se tramite l'utilizzo di Logo per i bambini viene facilitato lo sviluppo di concetti geometrici quali: angolo, dimensione dell'angolo e i relativi concetti geometrici. L'indagine ha riguardato dodici studenti di quarta primaria, sei hanno lavorato con Logo e sei avevano la funzione di controllo. Sono state effettuate tre interviste, una all'inizio, una a metà e una alla fine delle quaranta sessioni dell'esperienza; ogni sessione durava circa quaranta minuti e

durante queste lezioni i bambini hanno lavorato in coppia. Da queste interviste è emerso che solamente i bambini che avevano utilizzato Logo hanno mostrato progressi migliorando le loro nozioni, inizialmente solo intuitive, che sono diventate maggiormente matematiche, per quanto riguarda il concetto di angolo, della sua dimensione e dalla rotazione che compie. Anche per quanto riguarda le proprietà geometriche delle forme in modo particolare i bambini che hanno utilizzato Logo mostrano un miglioramento nella loro spiegazione. I bambini quindi diventano più consapevoli delle loro intuizioni matematiche e migliorano il loro pensiero geometrico.

Un ultimo studio che vorrei analizzare è quello di Cope e Simmons (1990) che si incentra sull'angolo. Molto interessante è la riflessione iniziale dei due autori che sottolinea come un insegnante non può ritenere che il semplice utilizzo libero di Logo, senza alcun supporto o preparazione dell'insegnante, può avere dei benefici sulla conoscenza dell'angolo e sulla sua misurazione. Dopo questa premessa lo studio ha cercato di analizzare due punti: se i bambini che hanno utilizzato Logo sanno descrivere l'angolo utilizzando i comandi del programma e come i bambini stimano le dimensioni degli angoli nel contesto di Logo. La ricerca ha coinvolto cinquantanove bambini tra i nove e i dodici anni, provenienti da due scuole diverse ma dove i bambini avevano utilizzato Logo. Dai risultati dei test è emerso che i bambini hanno capito la corrispondenza tra un determinato numero che viene scritto nei comandi e gli effetti sullo schermo ma solitamente associano questo effetto all'angolo interno invece che a quello esterno. Questa associazione porta poi ad avere difficoltà nella creazione del triangolo; quasi tutti i bambini hanno scritto correttamente il codice del quadrato, nel quale sia l'angolo interno che l'angolo esterno sono di novanta gradi, ma solamente un quarto dei bambini ha scritto correttamente il codice del triangolo. Gli autori sottolineano quindi che potrebbe essere importante cercare di lavorare sui concetti di angolo interno ed angolo esterno.

Gli studi sopra presentati sono stati fondamentali per strutturare le basi della sperimentazione descritta nel capitolo successivo.

Capitolo 3

Il progetto svolto in classe: coding con Logo e Scratch

3.1 La nascita del progetto

L'idea di questa sperimentazione è nata dopo aver frequentato il corso "Laboratorio di tecnologie didattiche" del professor Formiconi. Durante le prime lezioni il professore ci ha parlato del linguaggio di programmazione Logo, il nome non ha fatto nascere alcun ricordo in me, ma non appena ho visto la tartaruga, la mia mente mi ha riportato alle lezioni che avevo svolto durante la scuola primaria: tutti i tentativi, il divertimento e la soddisfazione quando riuscivo a far disegnare alla tartaruga le figure che volevo. Ho sempre avuto timore nel confronto degli errori, ma nonostante con Logo ci volessero molti tentativi e sbagli per riuscire ad ottenere un risultato, nei miei ricordi non è presente la paura di sbagliare o la tristezza per aver commesso degli errori. Ho deciso, quindi, di sperimentare immediatamente il programma di nuovo, mentre lo utilizzavo e grazie alle discussioni nate sul forum del corso, ho iniziato anche a notare la valenza educativa di questo software e ha riflettere su come avrei potuto utilizzarlo in classe. Durante il corso ho anche scoperto un altro linguaggio di programmazione "Scratch" che inizialmente mi ha creato un po' di difficoltà essendo un linguaggio visuale, mi sono domandata se questo potesse derivare dal fatto che il primo linguaggio che ho utilizzato è testuale o se ci fossero altre ragioni. Quando quindi ho saputo di avere la possibilità di svolgere il tirocinio in una classe 2.0 ho pensato che sarebbe stata un'occasione poter svolgere realmente delle lezioni con questi programmi, ho quindi incontrato il professor Formiconi e abbiamo pensato a questa sperimentazione. Ho poi parlato con la mia tutor scolastica la quale è stata molto disponibile e mi ha supportata durante lo svolgimento di questo percorso. Questo progetto però mi ha portata ad un'altra collaborazione, quella con Martina Fini una collega di corso interessata a svolgere una sperimentazione simile, parlandone insieme e sapendo della possibilità di svolgere tesi in collaborazione abbiamo deciso di proporre al professore di lavorare insieme. Entrambe sapevamo che una sola classe è un campione limitato

e quindi questa poteva essere un'occasione per ampliarlo, ma soprattutto ritenevamo entrambe importante il lavoro in team per un'insegnante che ci ha permesso di confrontarci per strutturare e svolgere al meglio queste lezioni, per la mia esperienza è stato molto interessante poter lavorare con un'altra collega e utile al fine di svolgere una sperimentazione migliore.

3.2 Descrizione del progetto

Il progetto si incentra sull'utilizzo del coding come sostegno nello studio della geometria, in particolare, abbiamo deciso di incentrarci sullo studio dell'angolo, dagli studi precedenti analizzati (ricerche analizzate nel capitolo 2) risulta che spesso i bambini hanno diverse misconcezioni sugli angoli (ad esempio la conservazione dell'angolo retto) e quindi abbiamo deciso di incentrarci su alcune di esse. Inoltre, è stato fatto un confronto tra due tipi di linguaggio di programmazione, uno visuale e l'altro testuale, per vedere se la diversa tipologia influenza l'apprendimento di questi concetti. I linguaggi utilizzati sono Logo, nella versione contenuta in LibreLogo e Scratch, nella versione online. La durata del progetto è stata differente nelle due classi, la classe quarta ha svolto le lezioni di coding per un totale di otto ore, ma ha anche svolto un laboratorio di scrittura legato al progetto dalla durata di tre ore. La classe quinta invece ha svolto le lezioni di coding per un totale di dodici ore e altre due ore sono state dedicate a una lezione che ha coinvolto i genitori dove i bambini hanno potuto provare a spiegare i due software utilizzati ai propri genitori. Sono stati somministrati anche dei test che hanno richiesto un tempo aggiuntivo di un'ora ciascuno: un pre-test (Test 1), un test intermedio (Test 2) somministrato a metà percorso e un test finale (Test 3). Insieme al test finale è stato somministrato un questionario che ha richiesto circa trenta minuti per la compilazione.

Le attività sono state svolte in aule attrezzate con una LIM e un computer per ogni coppia di bambini. Mentre un gruppo svolgeva le attività di coding, l'altro faceva lezione con la maestra che è anche la tutor scolastica che mi ha sostenuta durante il percorso.

3.3 I soggetti della sperimentazione

La sperimentazione ha riguardato due classi: una classe quinta primaria di Lucca, e una classe quarta primaria di Massa. Io ho svolto le mie lezioni nella classe quinta situata nell'Istituto Comprensivo Lucca 2 nel plesso "C.L. Collodi", la classe è composta da venti alunni e sono presenti due bambini DSA; la classe è una delle classi 2.0 e quindi ogni bambino ha la possibilità di utilizzare un computer personale. La classe dove ha svolto le lezioni Martina è, invece, una classe quarta dell'Istituto Comprensivo "Don Milani" di Massa, nel plesso "Vincenzo Giudice", la classe è composta da ventiquattro bambini, sono presenti un bambino con DSA e un bambino con sindrome di Asperger in attesa di certificazione. In totale quindi il campione è composto da quarantaquattro bambini è quindi sono consapevole che non sia molto ampio, ma può essere comunque una base per future ricerche. Ogni classe è stata suddivisa in due gruppi, un gruppo iniziava a programmare con Logo (*Logo-First*) mentre l'altro iniziava con Scratch (*Scratch-First*)²⁴ a metà percorso i due gruppi hanno cambiato linguaggio di programmazione.

3.4 Descrizione del progetto

Prima di iniziare lo svolgimento delle lezioni è stata necessaria un'attenta progettazione, di seguito riportata. Questo percorso si lega a tre differenti discipline: matematica, inglese e tecnologie. Per ogni disciplina sono stati individuati traguardi per lo sviluppo delle competenze e gli obiettivi di apprendimento, riprese dalle *Indicazioni Nazionali per il curricolo della scuola dell'infanzia e del primo ciclo d'istruzione* del 2012.

²⁴ Abbiamo scelto gli stessi nomi dei gruppi utilizzati dalla ricerca di Colleen Lewis (2010)

Disciplina: Matematica

Traguardi per lo sviluppo delle competenze relative alla disciplina Matematica:

- L'alunno rappresenta forme del piano.
- L'alunno descrive, denomina e classifica figure in base a caratteristiche geometriche, progetta e costruisce modelli concreti.
- L'alunno sviluppa un atteggiamento positivo rispetto alla matematica, attraverso esperienze significative.
- L'alunno sviluppa la capacità di comunicare e di discutere, di argomentare in modo corretto, di comprendere i punti di vista e le argomentazioni degli altri.

Obiettivi di apprendimento:

- Descrivere, denominare e classificare figure geometriche, identificando elementi significativi;
- Riprodurre una figura in base a una descrizione;
- Confrontare angoli.

Disciplina: Tecnologia

Traguardi per lo sviluppo delle competenze relativi alla disciplina Tecnologia:

- L'alunno produce semplici modelli o rappresentazioni grafiche utilizzando strumenti multimediali.

Obiettivi di apprendimento:

- Riconoscere e documentare le funzioni principali di una nuova applicazione informatica;
- Cercare e selezionare sul computer un comune programma di utilità.

Disciplina: Inglese

Traguardi per lo sviluppo delle competenze relativi alla disciplina Inglese:

- L'alunno comunica con espressioni memorizzate.

Obiettivi di apprendimento:

- Comprendere, leggere e scrivere le seguenti parole ed espressioni: “forward”, “back”, “right”, “left”, “home” e “clearscreen”.

Strumenti: Ogni coppia di bambini aveva a disposizione un computer, l'insegnante necessitava di un computer e della LIM. Affinché, ogni bambino potesse utilizzare i due software, ho installato su ogni computer LibreOffice e inserito la barra di Logo, questa operazione ha richiesto del tempo, ma si è rivelato vantaggioso avere un software che funzionava senza connessione. Per quanto riguarda Scratch è stato invece necessario preparare prima la classe virtuale creando un account per ogni coppia di bambino, usando il programma online, i bambini hanno dovuto inserire ogni volta l'account e ci sono stati piccoli problemi di connessione ma nel complesso non abbiamo avuto molte difficoltà; la possibilità di creare una classe su Scratch si è rivelata molto utile, ho potuto controllare i lavori svolti da casa anche senza doverli rendere pubblici. Sicuramente utilizzando i PC in diverse occasioni abbiamo dovuto fermare il lavoro per poter risolvere problemi tecnici, questo mi ha fatto riflettere sulla necessità di organizzare bene i tempi quando si utilizzano le tecnologie, anche se questo è valido per ogni tipo di lezione in realtà. Ho trovato diverse strategie per diminuire i tempi di preparazione, avendo avuto la possibilità di svolgere le lezioni durante le prime due ore ho potuto predisporre l'aula prima dell'inizio delle lezioni. Per prima cosa accendevo i computer controllando che non ci fossero problemi e sistemandoli nelle posizioni corrette, poi preparavo anche la LIM e il computer utilizzato da me. Avendo deciso di lavorare a coppie avevo dei computer in più sui quali ho comunque installato il software in caso uno dei computer avesse avuto qualche problema.

Tempi: il progetto è stato suddiviso in dodici lezioni e alcune ore sono state dedicate allo svolgimento dei test e del questionario:

- Somministrazione del pre-test (1 ora).

- *Lezione 1: Il primo incontro con Logo (1 ora).* L'attività è stata svolta dal gruppo *Logo-First* della classe quarta e quinta.
- *Lezione 2: Il primo incontro con Scratch (1 ora).* L'attività è stata svolta dal gruppo *Scratch-First* della classe quarta e quinta.
- *Lezione 3: Costruiamo alcune figure con Logo (1 ora).* L'attività è stata svolta dal gruppo *Logo-First* della classe quarta e quinta.
- *Lezione 4: Costruiamo alcune figure con Scratch (1 ora).* L'attività è stata svolta dal gruppo *Scratch-First* della classe quarta e quinta.
- *Lezione 5: Scopriamo un nuovo comando: il comando REPEAT di Logo (1 ora).* L'attività è stata svolta dal gruppo *Logo-First* della sola classe quinta.
- *Lezione 6: Scopriamo un nuovo comando: il comando RIPETI di Scratch (1 ora).* L'attività è stata svolta dal gruppo *Scratch-First* della sola classe quinta.
- Somministrazione test intermedio (1 ora).
- *Lezione 7: Il primo incontro con Logo (1 ora).* L'attività è stata svolta dal gruppo *Scratch-First* della classe quarta e quinta.
- *Lezione 8: Il primo incontro con Scratch (1 ora).* L'attività è stata svolta dal gruppo *Logo-First* della classe quarta e quinta.
- *Lezione 9: Costruiamo alcune figure con Logo (1 ora).* L'attività è stata svolta dal gruppo *Scratch-First* della classe quarta e quinta.
- *Lezione 10: Costruiamo alcune figure con Scratch (1 ora).* L'attività è stata svolta dal gruppo *Logo-First* della classe quarta e quinta.
- *Lezione 11: Scopriamo un nuovo comando: il comando REPEAT di Logo (1 ora).* L'attività è stata svolta dal gruppo *Scratch-First* della sola classe quinta.
- *Lezione 12: Scopriamo un nuovo comando: il comando RIPETI di Scratch (1 ora).* L'attività è stata svolta dal gruppo *Logo-First* della sola classe quinta.
- Somministrazione test finale (1 ora).
- Svolgimento del questionario (30 minuti).

Modalità di verifica: la verifica è stata svolta principalmente in itinere, infatti, durante l'utilizzo dei programmi ci sono stati diversi momenti di riflessione comune nei quali si cercava di riflettere su come risolvere possibili problemi, inoltre, mentre i bambini lavoravano in coppia, passavo tra i banchi per controllare il lavoro. Per quanto riguarda le conoscenze geometriche anche i test hanno permesso di verificare i risultati raggiunti.

3.5 Descrizione dell'esperienza

Le lezioni si sono svolte nella classe dove i bambini svolgono quotidianamente le attività didattiche essendo dotata di LIM e situata nel punto dove la connessione funziona al meglio. Le lezioni sono dodici in totale e si sono svolte nell'arco di un mese e mezzo, a distanza di una settimana l'una dall'altra. Nella stessa mattinata un'ora veniva dedicata al gruppo *Scratch-First* e l'altra al gruppo *Logo-First*.

Lezione 1: Il primo incontro con Logo

Quest'attività è stata svolta dal gruppo *Logo-First*, per la prima lezione i computer sono stati preparati ma non accesi poiché la prima parte della lezione non ne richiedeva l'uso e avrebbero potuto essere una fonte di distrazione. Per prima cosa ho spiegato ai bambini come posizionarsi e che avrebbero lavorato in coppia con un solo computer, abbiamo scelto di far lavorare i bambini con un compagno perché ritenevamo che in questo modo si sarebbero aiutati a vicenda e stimolati a trovare nuove soluzioni. Lavorando in coppia chi ha maggiori difficoltà può essere sostenuto dal compagno o può trovare un suo punto di forza e riuscire lui ad aiutare l'altro, ma è anche importante per chi è abituato a lavorare da solo dover imparare a condividere e ragionare insieme ad un'altra persona. Poi ho spiegato loro le attività che avremmo svolto e gli obiettivi che volevamo raggiungere. Prima dell'incontro i bambini avevano parlato di programmazione durante il "festival della didattica digitale" al quale abbiamo partecipato insieme, quindi, sono partita dalle loro preconoscenze e abbiamo cercato di riflettere insieme sul termine "coding". Siamo arrivati a capire che la programmazione è

legata alla scrittura di codici, allora ho detto loro che avrebbero imparato a scrivere dei codici per far muovere e disegnare una tartaruga e che, questo programma, ci avrebbe anche aiutato con la geometria; non appena hanno sentito il termine geometria in molti hanno reagito negativamente e questo mi ha portato a riflettere sul concetto di “mathophobia²⁵” di cui parlava Papert (1984). A questo punto ho chiesto ai bambini secondo loro che tipo di comandi possono far muovere la tartaruga, ho finto di essere la tartaruga e chiesto di darmi dei comandi per poter raggiungere un punto della classe; ho spiegato che la tartaruga si muove solamente se ha dei comandi precisi e che non mi sarei mossa nel caso in cui il comando non sarebbe stato chiaro per la tartaruga. I bambini hanno iniziato a fare delle ipotesi e quando non mi muovevo abbiamo insieme cercato di capire il perché. Il primo comando proposto è stato “vai avanti” e quando non mi sono mossa un altro bambino a provato ad inserire il numero di passi, a questo punto mi sono mossa, un altro alunno però mi ha fatto notare che il numero di passi in realtà dipendeva anche dalla loro lunghezza. Sono poi stati inseriti altri comandi, tra cui la rotazione e un alunno ha utilizzato i gradi per dirmi di quanto ruotare. Quest’attività ha permesso loro di iniziare ad avere confidenza con la programmazione e a capire che i comandi devono essere precisi. Ho poi detto agli alunni che la tartaruga capisce solamente l’inglese e quindi anche i comandi devono essere scritti in inglese, anche in questo caso qualcuno ha reagito negativamente, ma molti meno rispetto alla geometria, e alcuni hanno reagito con entusiasmo. Ho poi spiegato loro che il programma che avremmo utilizzato si chiama Logo e che dovevamo aprire LibreOffice per poterlo utilizzare, insieme abbiamo aperto la pagina per poter iniziare il lavoro in modo che capissero come farlo. A questo punto ho mostrato loro i quattro comandi principali (FORWARD, BACK, RIGHT e LEFT), spiegando che per farli eseguire dovevano cliccare il tasto PLAY sulla *toolbar*. Quando mostravo un comando nuovo davo ai bambini del tempo per provarlo; dopo averli mostrati tutti, hanno avuto alcuni minuti per usarli a loro piacimento. Mentre svolgevano quest’attività, girando tra i banchi, ho potuto ascoltare i loro ragionamenti e quando nasceva un problema diventava un

²⁵ Per Papert ha un doppio significato, è legato sia ai timori associati alla matematiche che diventano vere e proprie fobie, sia alla paura di imparare (“math” in greco significa apprendimento) un problema che nasce nel tempo poiché i bambini sono costretti ad imparare.

momento di discussione per tutti; un bambino ad esempio mi ha chiesto perché la tartaruga non eseguiva tutti i comandi, in realtà aveva fatto muovere la tartaruga a destra di 90° e poi a sinistra di 90° e quindi sembrava che non avesse fatto niente, è nato poi un problema alcuni volevano cancellare quello che avevano fatto. Partendo dal loro spunto ho spiegato altri due comandi che volevo introdurre: CLEARSCREEN, per cancellare quello che ha disegnato la tartaruga, e Home che riposiziona la tartaruga nel punto di partenza (il centro del foglio) e con la testa rivolta verso l'alto. L'ultima attività è stata quella di scrivere i comandi appresi su una tabella da me preparata che poi verrà inserita in un libricino che raccoglierà tutto il lavoro fatto. Questo lavoro è importante come punto di riferimento se vorranno continuare ad utilizzarli dopo aver finito le attività, ma anche perché durante il lavoro è sicuramente utile avere anche un supporto scritto.

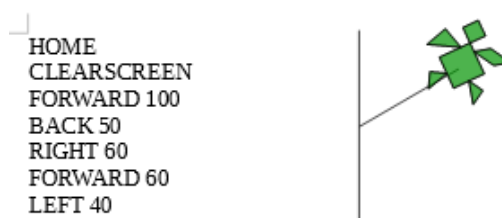


Figura 10 Uno dei codici scritti da una coppia di bambini.

Lezione 2: Il primo incontro con Scratch

Quest'attività è stata svolta dal gruppo *Scratch-First*. Anche questa lezione è iniziata cercando di attivare le preconcoscienze dei bambini sulla programmazione per cercare di capire cos'è insieme, poi ho esplicitato gli obiettivi che volevamo raggiungere: imparare le basi della programmazione e farci aiutare con la geometria. Questo gruppo non ha reagito negativamente quando ha sentito il termine geometria come era successo con il primo gruppo, ma è stato anche meno partecipe nella discussione per individuare cos'è il coding. Ho poi spiegato loro che avremmo utilizzato il programma Scratch per far muovere e disegnare un gatto; mi sono proposta come finto gatto al quale dare i comandi per raggiungere un punto dell'aula. I bambini anche in questa lezione hanno formulato molte ipotesi sui comandi che inizialmente erano imprecisi, “cammina” o “gira”, ma

vedendo che i miei movimenti non portavano al risultato continuavano a fare ipotesi finché non hanno capito che ad esempio oltre al comando “gira” è necessario dire la direzione e i gradi. In questo modo hanno avuto la possibilità di sperimentare che se il comando non è preciso il gatto non eseguirà quello che si era preventivato e hanno avuto modo di capire i comandi principali che potranno servirgli. Dopo, abbiamo insieme aperto Scratch in modo che ogni bambino vedesse come fare e ogni coppia ha inserito il nome utente e la password che avevo preparato, quest’operazione ha richiesto diverso tempo sia per poter inserire l’account sia perché abbiamo dovuto aspettare che la pagina si caricasse per tutti. Insieme abbiamo visto come si presentava il programma, alcuni bambini lo conoscevano già e questo ha portato alla richiesta di cambiare lo *sprite*, ma ho spiegato la necessità prima di capire come funziona il programma e quindi abbiamo continuato, ho presentato la sezione dei blocchi e poi ho chiesto ai bambini secondo loro dove potevamo trovare i comandi per farlo muovere e ho illustrato alcuni blocchi principali “fai ... passi” e “ruota di ...” nelle due direzioni, mostrandoli e facendoli provare ai bambini, poi ho detto nuovamente che dovevamo far disegnare la tartaruga e quindi hanno cercato un comando che potesse svolgere questa funzione e abbiamo quindi utilizzato “penna giù”. Ho poi mostrato come far eseguire i comandi in maniera diversa, fino a quel momento avevano cliccato sul comando da eseguire ma poi ho spiegato come incastrare i blocchi e che potevano utilizzare “quando si clicca su bandierina” per far eseguire tutti i comandi semplicemente premendo la bandierina verde sulla parte alta dello stage, poi abbiamo visto anche come eliminare i comandi inseriti. Ho anche introdotto altri tre comandi “pulisci”, “vai a x 0 e y 0” e “punta in direzione ...” spiegando loro le funzioni, ma dicendo che avremmo capito bene come usarli nella prossima lezione. Dopo li ho lasciati provare il programma, anche perché in molti avevano già iniziato a inserire alcuni blocchi, sicuramente rispetto alla prima lezione è stato più difficile mantenere l’attenzione durante l’uso del programma, perché volevano provare subito i blocchi ed essendo già pronti era più facile distrarsi. Durante queste lezioni spesso nascono molti scambi tra i bambini ed entusiasmo creando la cosiddetta “confusione” ma non deve essere vista negativamente, durante queste attività è fondamentale potersi confrontare

con gli altri e discutere strategie. Come ultima attività abbiamo creato anche per Scratch una lista dei comandi da inserire poi nel quadernino che conterrà i lavori fatti; in questo caso invece di scrivere i comandi ogni bambino aveva dei blocchi da incollare sulla parte sinistra del foglio mentre nella parte destra abbiamo insieme trovato delle spiegazioni per ogni blocco in modo che per loro fossero chiare.



Figura 11 Blocchi utilizzati nella prima lezione dai bambini

Lezione 3: Costruiamo alcune figure con Logo.

Quest'attività è stata svolta del gruppo *Logo-First*. La lezione riparte da ciò che è stato appreso nella lezione precedente, in questo modo possiamo osservare cosa hanno realmente appreso, ripetere le cose in modo che chi non ha compreso appieno possa farlo e rispiegare per gli assenti. Quindi ho chiesto ai bambini di spiegare a turno i comandi che abbiamo introdotto nella precedente lezione e poi, ho distribuito la lista di comandi scritta da loro per poterli avere come supporto durante il lavoro. Questa lezione si incentra particolarmente sulla geometria e quindi ho chiesto loro di provare a disegnare un quadrato, prima abbiamo cercato di individuare le caratteristiche di questa figura e subito mi hanno detto che ha quattro lati uguali e quattro angoli uguali di 90° . Poi, hanno lavorato in coppia per cercare di scrivere il codice per disegnare un quadrato, interessante è stato osservare che hanno creato la figura con tempi diversi ma soprattutto con metodi diversi, una coppia invece di far andare avanti la tartaruga ha deciso di farla disegnare con il comando BACK, la maggior parte delle coppie ha disegnato un quadrato. Quando stava per terminare la nostra ora, non potendo disporre di tempo

aggiuntivo, ho deciso di far venire alcune coppie a spiegare il loro ragionamento in modo che chi non era riuscito a disegnare il quadrato potesse capire come fare, insieme abbiamo ragionato sul fatto che per creare la figura andava bene qualsiasi misura dei lati ma che gli angoli dovevano essere sempre di 90° . Insieme abbiamo anche ragionato sulla direzione della tartaruga e notato che, dopo aver finito di fare il quadrato puntava nuovamente nella direzione iniziale perché in realtà aveva ruotato di 360° . Ho poi chiesto loro di provare a disegnare un triangolo equilatero e anche in questo caso abbiamo evidenziato le sue caratteristiche. Per questa figura le difficoltà sono risultate maggiori, infatti, come avevamo preventivato i bambini hanno dato per scontato che inserendo il valore dell'angolo interno del triangolo la figura sarebbe stata quella immaginata; questo errore è molto comune anche Cope e Simmons (1990) hanno evidenziato nella loro ricerca che sia un aspetto su cui lavorare, io stessa durante le mie prime sperimentazioni con Logo ho effettuato lo stesso errore. La figura in questo modo risulta così

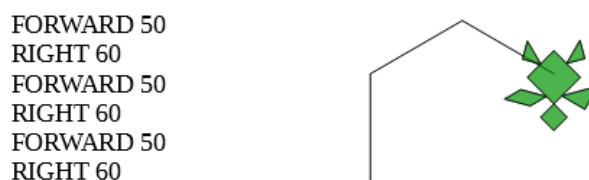


Figura 12 Prova creazione triangolo equilatero

Insieme abbiamo quindi riflettuto su questa problematica e ho cercato di spiegare loro perché succedeva così e che dovevano calcolare l'angolo esterno e non quello interno. In questo modo hanno poi potuto costruire un triangolo equilatero ma è stato un passaggio complesso da capire per molti di loro. Dopo ho introdotto alcuni nuovi comandi PENCOLOR, FILLCOLOR, PENUP e PENDOWN e ho chiesto agli alunni di provare a creare una casetta colorata. Sono stati molto soddisfatti del loro lavoro, una coppia ha anche cercato di disegnare una porta e alcuni dagli "errori", come ad esempio non chiudere completamente la figura, hanno creato delle casette molto originali. Insieme abbiamo riflettuto sul perché siano venute diverse da come avevano immaginato, ma non sono state etichettate come qualcosa di sbagliato anzi sono state valorizzate. Infine abbiamo aggiunto i

nuovi comandi nella nostra lista.

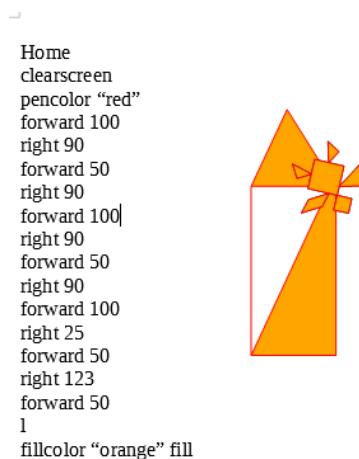


Figura 13 Una delle case con “errore” creata dai bambini

Lezione 4: Costruiamo alcune figure con Scratch

Quest’attività è stata svolta del gruppo *Scratch-First*. La lezione è iniziata riprendendo i comandi presentati la volta precedente e consegnando la lista creata dai bambini. Ho poi chiesto loro di utilizzare i blocchi per scrivere un codice che porti il gatto a disegnare un quadrato. Insieme abbiamo ripreso le caratteristiche della figura e poi in coppia hanno cercato di riprodurla. Anche per questo gruppo sono state utilizzate strategie diverse, con Scratch qualcuno ha scritto solamente due comandi “fai 100 passi” e “ruota di 90°” e li hanno fatti eseguire quattro volte. Ho quindi chiesto loro, nonostante fosse un metodo efficace, di provare a scrivere il codice completo poi, alcune coppie sono venute a esporre il metodo che avevano utilizzato e abbiamo riflettuto sul fatto che nonostante i lati fossero di dimensioni diverse gli angoli dovevano essere sempre di 90° in questo modo abbiamo anche sottolineato che la lunghezza del lato non influenza le dimensioni dell’angolo. Poi ho chiesto loro di provare a disegnare un triangolo e anche questo gruppo ha avuto le stesse difficoltà di quello precedente per questa figura, un bambino ha detto “il gattino ha sbagliato, non credo che abbia capito cosa fare” e, quindi, ha provato a mostrare alla lavagna perché il gatto ha fatto così. Partendo dal suo ragionamento abbiamo riflettuto insieme sul problema e altri bambini hanno mostrato le loro ipotesi finché non è stata trovata una soluzione, i bambini

hanno capito che è necessario scrivere la misura dell'angolo esterno in modo che l'angolo interno risulti dell'ampiezza desiderata. Si sono domandati perché per il quadrato sia diverso e hanno capito che in realtà il quadrato ha l'angolo esterno e quello interno delle stesse dimensioni. Ho poi introdotto i comandi “usa penna di colore”, “cambia colore penna di ...” e “usa penna di dimensione”; ho chiesto di usare anche i nuovi comandi e provare a disegnare la lettera iniziale del loro nome, non usando però linee curve, quest'attività è stata molto più complicata di quello che avevo preventivato e la maggior parte dei bambini non ha terminato la figura. Poi abbiamo inserito i nuovi comandi appresi nella lista.

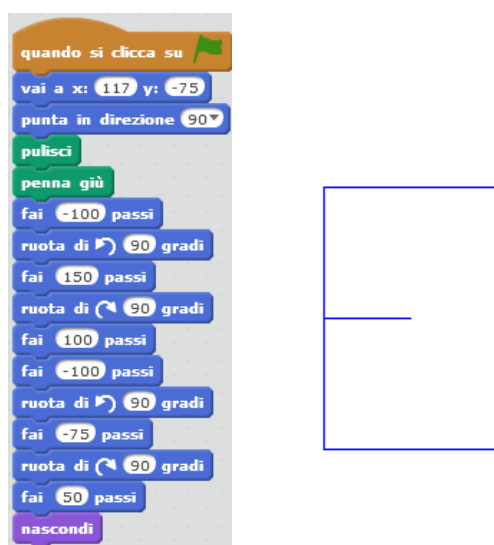


Figura 14 Lettera iniziale creata da un'alunna

Lezione 5: Scopriamo un nuovo comando: il comando REPEAT di Logo

Quest'attività è stata svolta del gruppo *Logo-First*. Questa lezione è stata svolta solamente dalla classe quinta. Come per l'altra lezione riprendiamo quello che abbiamo fatto nella lezione precedente e distribuisco i fogli con i comandi scritti da loro. Mostro il codice di un quadrato che hanno creato e domando qual è la prima cosa che notano osservando questo codice per arrivare a capire che gli stessi

comandi sono stati ripetuti quattro volte e che quindi servirebbe un unico comando per poter far ripetere le azioni alla tartaruga. Introduco quindi il REPEAT, come per le virgolette con i comandi per colorare la penna e le figure spesso i bambini commettevano errori a causa di uno spazio in più o una parentesi non chiusa correttamente, questo è uno degli aspetti che in parte complica il lavoro con questo tipo di linguaggio.

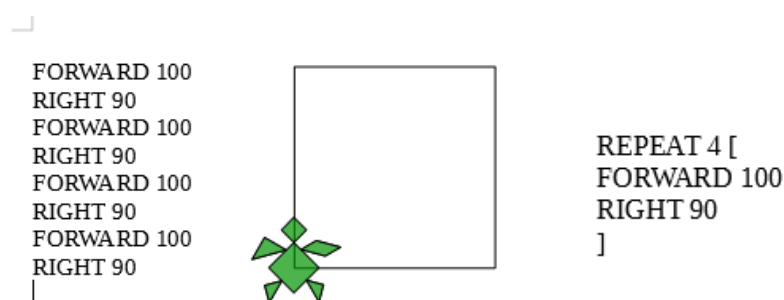


Figura 15 Il codice del quadrato con e senza REPEAT

Successivamente viene chiesto ai bambini di disegnare altri poligoni regolari a loro scelta cambiando il colore della penna e colorando la figura. Durante quest'attività in molti continuavano a scrivere l'ampiezza sbagliata dell'angolo per questo motivo ho spiegato nuovamente che devono prendere in considerazione l'angolo esterno, questi tentativi hanno portato anche in questo caso alla creazione di figure diverse ma che sono piaciute molto ai bambini. Infine abbiamo riflettuto su cosa hanno appreso e ho sottolineato come quello che abbiamo fatto sia solo una piccola parte di quello che è possibile creare con Logo.

Lezione 6: Scopriamo un nuovo comando: il comando RIPETI di Scratch

Quest'attività è stata svolta del gruppo *Scratch-First*. Questa lezione è stata svolta solo dalla classe quinta. La lezione inizia riprendendo quanto appreso la volta scorsa e riconsegnando i comandi. I bambini hanno poi terminato il lavoro della scrittura della lettera iniziale, ho capito che spesso il problema riguardava l'ampiezza dell'angolo e, quindi, ho ripetuto la spiegazione su come far ruotare il gatto affinché l'angolo sia quello desiderato. Poi ho mostrato il codice di un

quadrato e ho chiesto ai bambini se notavano qualcosa in particolare, anche in questo caso, con il mio supporto, hanno notato che il codice si ripeteva e ho poi introdotto il blocco “ripeti ... volte”.

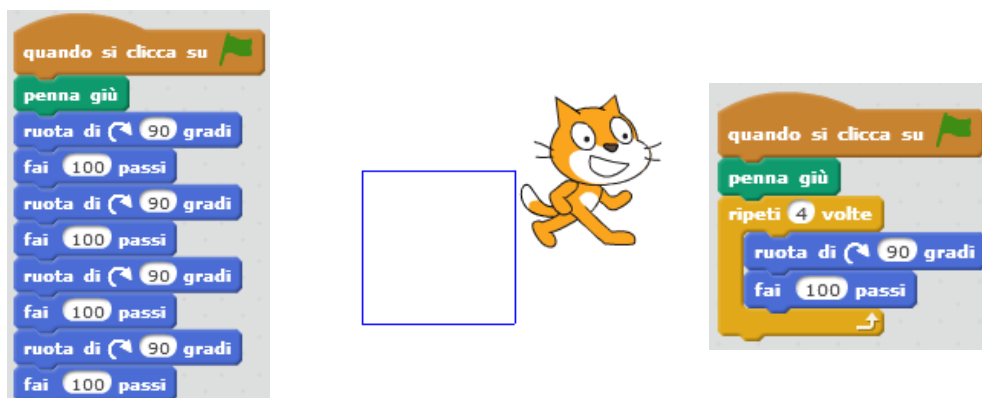


Figura 15 Il codice del quadrato con e senza il REPEAT

Ho poi chiesto loro di disegnare un altro poligono regolare. Infine abbiamo riflettuto insieme su quello che hanno imparato e poi ho spiegato che con Scratch è possibile creare moltissime altre cose e che, se vorranno, potranno provare ad utilizzarlo in maniera differente.

Dopo queste lezioni è stato svolto il test intermedio.

Le lezioni descritte in seguito riprendono il lavoro fatto in quelle sopra analizzate, ma i due gruppi hanno cambiato linguaggio di programmazione, chi usava Logo usa Scratch e viceversa. Non descriverò quindi cosa è stato fatto nei dettagli ma mi soffermerò sulle riflessioni più interessanti, poiché, usando i programmi dopo aver già conosciuto gli altri, sono nati pensieri diversi e comunque ogni gruppo di bambini è diverso e quindi non è possibile ottenere gli stessi feedback. Sicuramente molti concetti erano già stati appresi e questo a permesso di dedicare più tempo alla programmazione libera.

Lezione 7: Primo incontro con Logo

Quest'attività è stata svolta dal gruppo *Scratch-First*. Trovarsi di fronte ad un foglio bianco, a dover scrivere il codice e doverlo anche scrivere in inglese, inizialmente, è sembrato difficile ai bambini e meno stimolante. Ho allora cercato di riprendere i comandi che avevano utilizzato con Scratch e ho introdotto quelli nuovi e poi ho deciso di chiedere loro di provare a scrivere il codice per fare un quadrato e a questo punto sono stati maggiormente coinvolti perché hanno dovuto ragionare su come poterlo fare con comandi diversi e ci sono riusciti, seppur con iniziali incertezze. Quando ho chiesto di individuare le differenze, i bambini hanno quindi subito indicato il dover scrivere il codice e che la prima cosa che appare è solamente un foglio bianco.

Lezione 8: Il primo incontro con Scratch

Quest'attività è stata svolta del gruppo *Logo-First*. Quando abbiamo aperto la schermata principale, il primo commento di uno degli alunni è stato “sembra un gioco!” e molti di loro sono rimasti colpiti da quanto fosse diverso rispetto a Logo. Chi lo conosceva voleva utilizzarlo in maniera diversa, ma ho spiegato che avremmo cercato di fare le cose che avevamo fatto fare alla tartaruga con il gattino. Anche per questo gruppo per spiegare i comandi sono partita del far dire loro i comandi di Logo e poi ho chiesto di individuare i blocchi che secondo loro avevano la stessa funzione. Durante la presentazione dei blocchi di movimento una bambina mi ha fatto notare che il gattino non scriveva come la tartaruga ma che aveva visto dei blocchi nominati “penna” e quindi ho introdotto anche “penna giù”. Poi, ho chiesto che differenze notavano e la prima cosa che è stata evidenziata è che non dovevano scrivere i comandi. Avendo già appreso le basi molti bambini volevano provare subito il programma ed erano curiosi di creare qualcosa e quindi ho anticipato la creazione del quadrato anche con questo gruppo; tutte le coppie sono riuscite a farlo, ma un bambino mi ha detto “va troppo veloce non si vede quello che fa” questa è una caratteristica di Scratch che

avevo notato anch'io e che non permette al bambino di percepire il tipo di movimento che lo *sprite* fa.

Lezione 9: Costruiamo alcune figure con Logo

Quest'attività è stata svolta dal gruppo *Scratch-First*. La creazione del triangolo nonostante fosse già stata fatta con Scratch ha creato difficoltà ad alcune coppie, ma poi ragionando insieme in poco tempo tutti hanno svolto la figura. I nuovi comandi invece, nonostante siano stati accolti con entusiasmo, inizialmente sono risultati difficili poiché spesso nascevano errori di sintassi, ma, anche in questo caso, dopo alcune prove sono riusciti a creare una casetta colorata come avevo richiesto; anche in questo gruppo dagli "errori" sono nate diverse figure interessanti che hanno dato molta soddisfazione ai bambini.

Lezione 10: Costruiamo alcune figure con Scratch

Quest'attività è stata svolta del gruppo *Logo-First*. Nella creazione del triangolo non ci sono stati particolari problemi, anche se una coppia inizialmente aveva difficoltà. In questo caso le difficoltà sono nate maggiormente per quanto riguarda il posizionare lo *sprite* come volevano e quindi la scrittura della loro lettera iniziale ha comunque richiesto molto tempo e ha creato difficoltà, ma al contrario del primo gruppo hanno tutti terminato questo compito. I nuovi comandi sono subito stati compresi ed utilizzati efficacemente.

Lezione 11: Scopriamo un nuovo comando: il comando REPEAT di Logo

Quest'attività è stata svolta dal gruppo *Scratch-First*. Il nuovo comando è stato compreso e utilizzato immediatamente. I bambini hanno quindi avuto più tempo per creare altri poligoni regolari e infine abbiamo terminato i comandi da inserire nel libretto controllando che tutti avessero finito il lavoro.

Lezione 12: Scopriamo un nuovo comando: il comando RIPETI di Scratch

Quest'attività è stata svolta del gruppo *Logo-First*. Anche questo gruppo ha immediatamente compreso il nuovo comando e creato poligoni regolari e anche altre figure. Abbiamo infine terminato di scrivere la lista dei comandi.

Al termine delle lezioni, dopo la somministrazione dell'ultimo test, alcuni bambini mi hanno chiesto come poter continuare ad utilizzare il programma, avendo a disposizione una pagina su Moodle della classe, ho inserito tutte le istruzioni per scaricare LibreOffice e i dati per continuare ad utilizzare i loro account Scratch.

3.6 Presentazione dei test di geometria

Per poter acquisire dati sulle conoscenze dei bambini e verificare gli effetti delle lezioni sulle loro competenze sono stati svolti tre test: il primo (Test 1) è un pre-test che è stato somministrato prima dell'inizio delle lezioni; il secondo (Test 2) è stato svolto a metà percorso prima dello scambio di linguaggi di programmazione tra i due gruppi e l'ultimo (Test 3) è stato somministrato al termine di tutte le attività. Solo il Test 2 permette di vedere al meglio la differenza tra i due linguaggi ma il Test 3 svolge un ruolo importante per quanto riguarda l'acquisizione dei concetti geometrici; i bambini, infatti, hanno avuto modo di rafforzare i concetti acquisiti nella prima parte del percorso, non solo ripetendo quello che avevano imparato, ma anche utilizzando un linguaggio diverso che potrebbe essere maggiormente efficace per loro tenendo in considerazione che ogni bambino apprende in maniera differente.

I tre test sono stati preparati riprendendo alcuni degli *items* utilizzati nelle ricerche di Clemens e Battista (1990) e da Noss (1987).

I tre test sono strutturati in maniera molto simile tra loro per non rischiare che i risultati sbagliati siano legati ad una maggiore difficoltà del test. Sono state quindi modificate leggermente le figure o cambiati i numeri, ma ogni esercizio riguardava lo stesso argomento di quello corrispondente nel test precedente (ad

esempio per il riconoscimento dell'angolo retto, anche se non posizionato nel modo standard con il quale solitamente viene rappresentato sono stati presentati angoli retti posizionati diversamente tra di loro). Quando però abbiamo analizzato il Test 2 per entrambe le classi abbiamo notato che spesso i bambini pensando che l'esercizio fosse uguale al test precedente hanno risposto erroneamente, probabilmente poiché non avevano letto bene il testo e, quindi, abbiamo deciso di modificare maggiormente alcune domande del Test 3. Sicuramente la lettura sbagliata della domanda ha portato a diversi errori nei test poiché spesso i bambini hanno cercato di consegnare il prima possibile, nonostante li avessi rassicurati sul fatto che non era importante consegnare in fretta, infatti, quando dopo i tre test abbiamo analizzato insieme le domande molti bambini mi hanno saputo rispondere, anche se nel test avevano sbagliato alcune domande e mi hanno detto che avevano letto male la consegna o che per loro non era stata chiara. Allego nell'appendice i tre test somministrati.

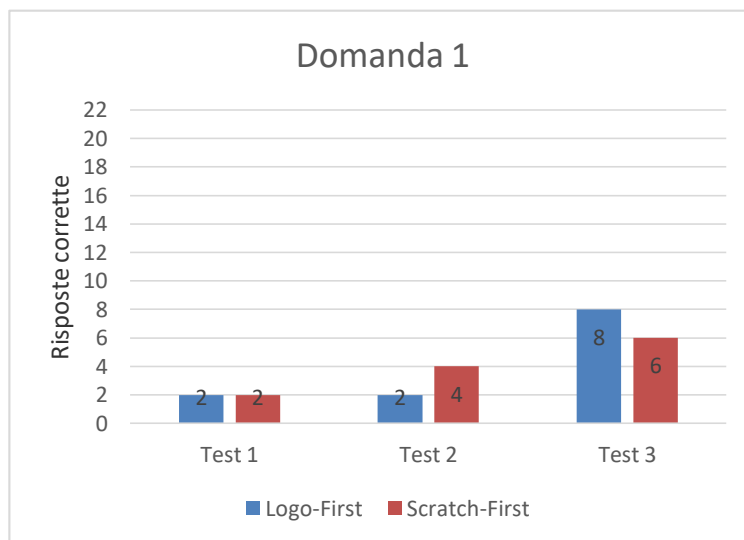
3.7 Risultati del test

Ogni test aveva la durata di un'ora, ma molti bambini hanno consegnato il test prima della scadenza, la maggior parte nella prima mezz'ora. Durante il primo test tutti e quarantaquattro bambini erano presenti, al secondo test invece non erano presenti tre bambini (uno del gruppo *Scratch-First* e due del gruppo *Logo-First*) e durante l'ultimo test vi erano due assenti (uno del gruppo *Scratch-First* e uno del gruppo *Logo-First*).

Di seguito vengono riportate tutte le domande presenti nei tre diversi test e i rispettivi risultati rappresentati con dei grafici che mostrano i numeri delle risposte corrette, dividendo i risultati per i due gruppi di trattamento (*Logo-First* e *Scratch-First*).

Domanda 1

Test 1 – 2 – 3: Cos'è un angolo? Disegnane uno.



Nel Test 1 le risposte alla domanda “Che cos'è un angolo?” sono state le seguenti:

L'angolo è...

lo spazio che sta tra due linee che si incontrano; lo spazio tra due linee; lo spazio tra due segmenti; lo spazio geometrico tra due lati consecutivi; lo spazio che calcola l'interno di due semirette spezzate e incrociate; un piano formato da due linee che si uniscono; una forma che collega i due lati e può essere di varie misure; la parte di piano compresa tra due segmenti; una linea spezzata che si incontra con i vertici; due linee che si incontrano; due semirette che si incontrano; due segmenti che si incontrano; due semirette che partono dallo stesso punto; il punto dove si incontrano due semirette; il punto dove si incontrano due segmenti; il punto dove si incontrano due lati; l'angolo retto fatto per i triangoli e altri; la distanza tra due segmenti, che si specifica in gradi; una cavità tra due linee; permette di misurare i gradi della figura, è una specie di misurazione, però non si misura con il righello ma con il goniometro; è un arco che deve stare sopra un vertice.

Nel Test 2 le risposte alla domanda “Che cos’è un angolo?” sono state le seguenti:

L’angolo è...

lo spazio che sta tra due segmenti che iniziano nello stesso punto; spazio che sta tra due semirette; parte di piano; parte di piano in cui si incontrano due semirette che si andranno ad unire; parte di piano compresa tra due segmenti che partono dallo stesso punto; due segmenti che iniziano e finiscono nello stesso punto; due semirette che si incontrano nello stesso punto; due semirette che partono dallo stesso punto; lo spazio che si trova fra 2 segmenti; una linea spezzata semplice chiusa; una semiretta e un incrocio tra semirette; un incrocio tra due segmenti; un incrocio tra due semirette; punto dove due lati si incontrano; l’interno di un vertice contato in gradi; incrocio di due semirette che nascono dallo stesso punto; si crea dall’incontro di due lati; punto di incontro tra due segmenti ed è quello spazio; punto in cui due lati si incontrano; il punto in cui due segmenti si incontrano; punto di incontro tra due linee che si forma dentro una figura; spazio compreso tra due segmenti che partono dallo stesso punto; una linea spezzata semplice aperta; una cavità tra due linee ma quando l’angolo è giro la cavità non c’è; una parte che indica l’ampiezza dell’immagine geometrica; un punto di una figura dove due segmenti combaciano tra loro; si esprime in gradi; è formato da due angoli che si incrociano perpendicolarmente o non perpendicolarmente formando un angolo; una linea con un'altra che si attaccano; un segmento con l’ampiezza; quando due rette o semirette si incontrano formando una x o molti altri simboli come una L o una A; un angolo è una parte fondamentale per sapere se un triangolo è retto....; un angolo retto aperto o chiuso; un arco sui lati.

Nel Test 3 le risposte alla domanda “Che cos’è un angolo?” sono state le seguenti:

L’angolo è...

parte di piano compresa tra due semirette; parte di piano compresa tra due segmenti; parte di piano in cui si incontrano due segmenti che nascono da uno stesso punto; spazio compreso in un piano formato dall’incontro tra due segmenti che nascono dallo stesso punto; parte di piano compreso tra

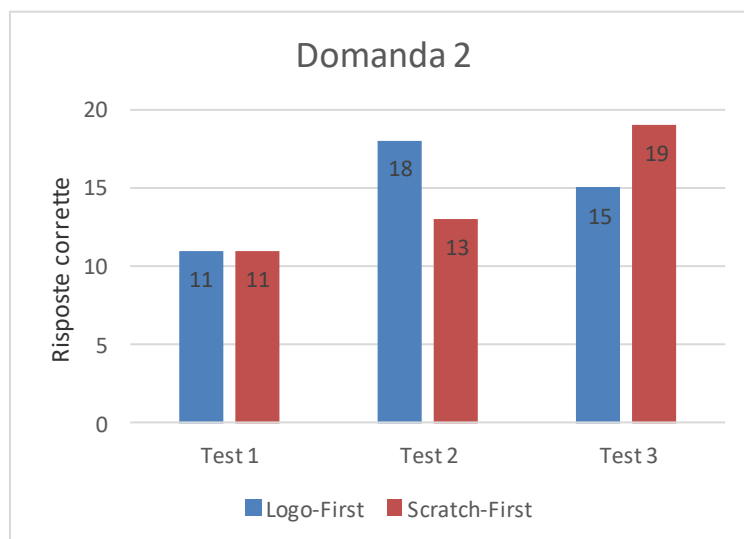
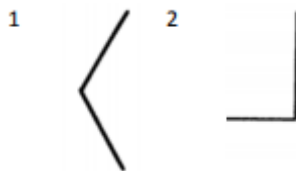
due segmenti che partono dallo stesso punto; spazio compreso tra due semirette che nascono dallo stesso punto; spazio che si trova fra due segmenti che si incontrano al vertice; parte di piano compreso tra due semirette che nascono dallo stesso punto; parte compresa tra due semirette (ampiezza) che partono dallo stesso punto; parte di piano; spazio compreso tra due semirette; spazio fra due lati; area tra due rette che si incontrano unione tra due linee semirette; vertice in cui due lati si incontrano; un incrocio tra due semirette; interno di due semirette nato dallo stesso punto; punto in cui un segmento nasce o finisce; incontro tra due semirette; punto in cui si incontrano due segmenti; parte compresa tra due lati; punto di incontro tra due semirette; quando due semirette si incrociano formando una rientranza; una linea aperta semplice chiusa; ampiezza che formano due semirette incrociate; archetto che dice la distanza tra due lati; punto di riferimento che indica l'ampiezza dell'immagine; distanza tra due semirette e si calcola in gradi

Dalle risposte aperte possiamo osservare come il concetto di angolo non sia chiaro prima dell'inizio delle lezioni, ma anche dopo l'utilizzo di Scratch e Logo, nonostante ci siano stati dei miglioramenti, specialmente nell'ultimo test, ancora non sia un concetto chiaro per la maggioranza dei bambini. Molti, infatti, ritengono che siano due segmenti che si incontrano o che il vertice rappresenti l'angolo, manca quindi la concezione dello spazio contenuto tra le due semirette che hanno origine nello stesso punto, che solitamente non viene considerato dai bambini.

Domanda 2

Test 1 – 2: Puoi disegnare un angolo maggiore rispetto a quello che hai disegnato prima? Perché è maggiore?

Test 3: Quale tra i due angoli è il maggiore? Perché?



In tutti e tre i test i bambini hanno identificato correttamente l'angolo maggiore, però molti non hanno saputo indicare il perché. Potrebbe esserci una correlazione con il fatto che i bambini non hanno chiara la definizione di angolo e quindi non hanno saputo spiegare perché risultava maggiore l'angolo selezionato. Un dato da osservare è che nel secondo test sono migliorati i risultati di entrambi i gruppi mentre nell'ultimo test i bambini del gruppo *Logo-First*, dopo la seconda parte di lezioni nella quale hanno usato Scratch, sono peggiorati mentre quelli di *Scratch-First*, che hanno invece iniziato ad usare Logo, sono migliorati.

Domanda 3

Test 1: Cerchia tra queste figure quelle che rappresentano angoli.

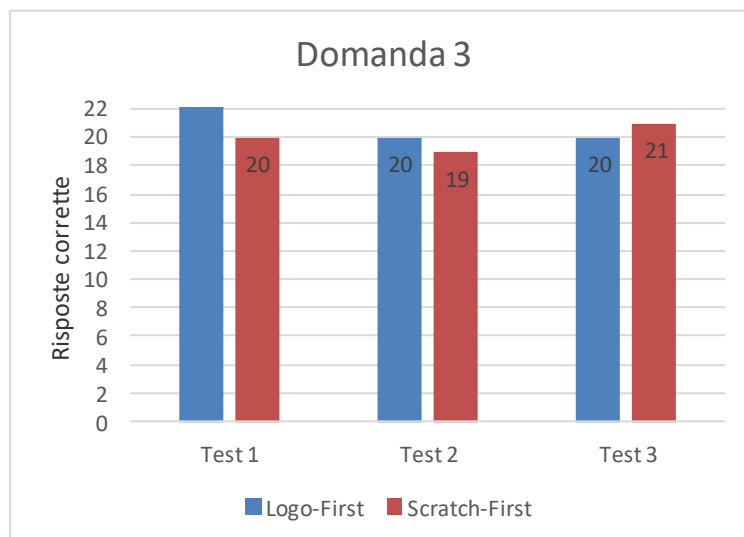


Test 2: Cerchia tra queste figure quelle che rappresentano angoli.



Test 3:



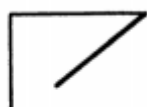


Nonostante piccole variazioni i risultati in tutti i test sono stati positivi. I bambini hanno individuato correttamente quali figure rappresentano gli angoli e alcuni hanno spesso segnalato i segmenti come angoli piatti, disegnando degli archetti per far capire che si trattava di un angolo.

Domanda 4

Quanti angoli hanno queste figure? Scrivi il numero di angoli sotto ad ogni figura.

Test 1:

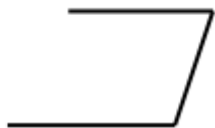


..... Angoli

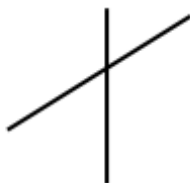


..... Angoli

Test 2:

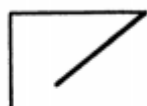


..... Angoli



..... Angoli

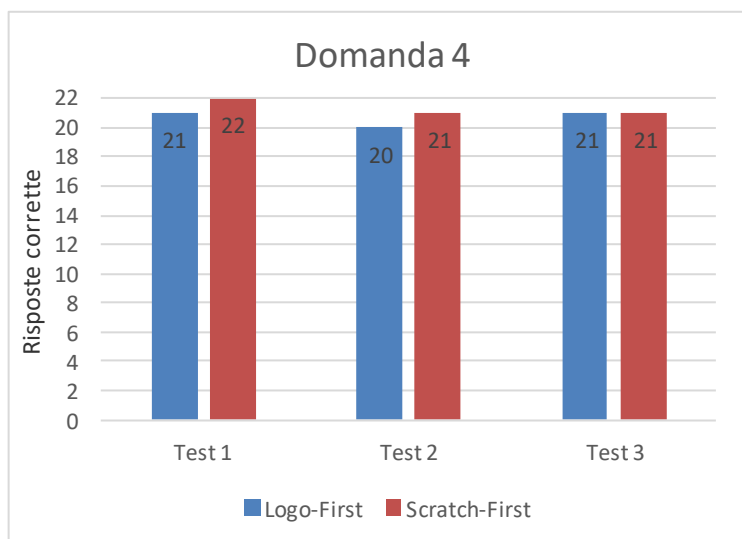
Test 3:



..... Angoli



..... Angoli

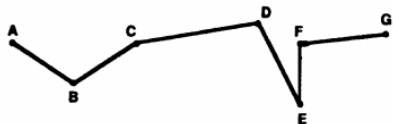


Anche le risposte di questa domanda non hanno subito grandi variazioni e sono state tendenzialmente positive. Per le prime figure di tutti e tre i test sono state considerate corrette le risposte che indicavano anche solamente gli angoli interni, ma alcuni bambini hanno segnalato anche gli angoli esterni.

Domanda 5

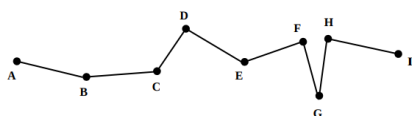
Test 1:

Stai camminando su questo sentiero. Inizi dal punto A e finisci nel punto G. In quale punto devi ruotare di più? Indicalo con una freccia di colore blu. In quale punto devi ruotare di meno? Indicalo con una freccia di colore rosso.



Test 2:

Stai camminando su questo sentiero. Inizi dal punto A e finisci nel punto I. In quale punto devi ruotare di più? Indicalo con una freccia di colore blu. In quale punto devi ruotare di meno? Indicalo con una freccia di colore rosso.

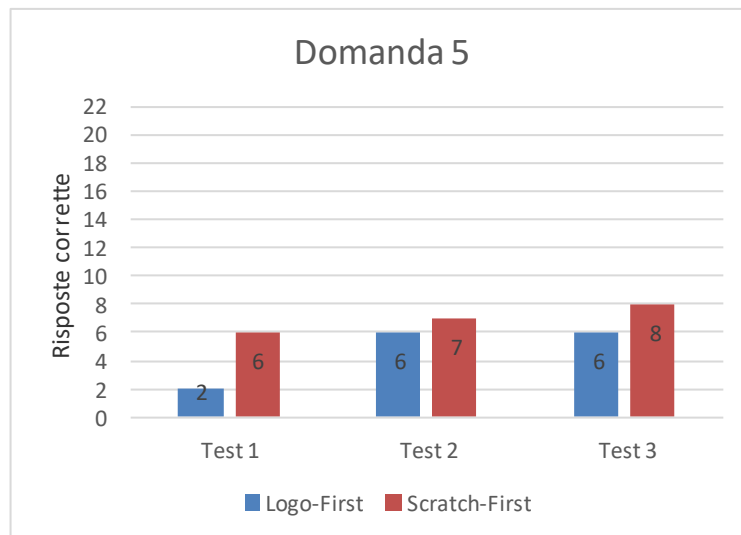


Test 3:

Adesso prova tu a disegnare un sentiero che inizi nel punto A e finisca nel punto G. Ad ogni cambio di direzione scrivi una lettera. Indica con una freccia di colore blu il punto in cui devi ruotare di più e con una freccia di colore rosso il punto in cui devi ruotare di meno.

A.

.G



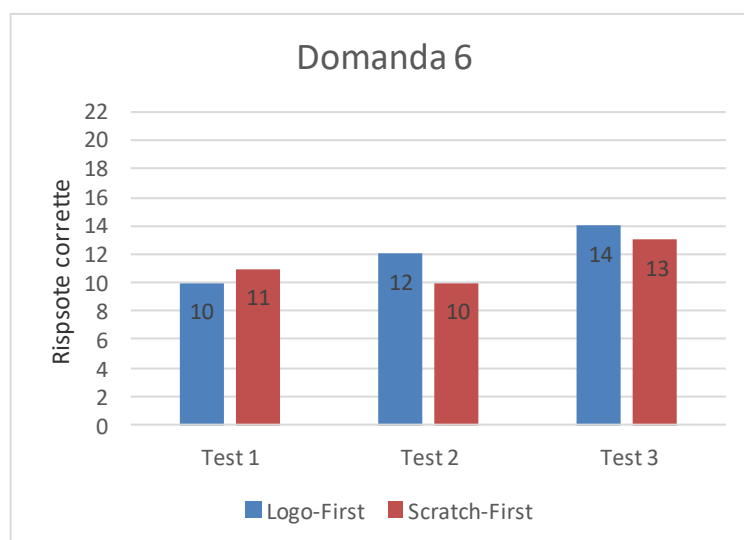
Questa domanda ha invece creato difficoltà alla maggioranza dei bambini. Le risposte corrette risultano poche rispetto alla totalità del campione. Abbiamo deciso di ritenere corrette solamente le risposte di chi ha individuato sia il punto minore di rotazione sia quello maggiore. Guardando le singole risposte, i bambini hanno individuato con più facilità il punto in cui doveva ruotare di più rispetto al punto in cui dovevano ruotare di meno. Interessante osservare le risposte di alcuni bambini che hanno invertito i due punti. Discutendo alla fine del lavoro con loro mi hanno spiegato che avevano considerato l'angolo con ampiezza maggiore e quello con ampiezza minore, non quello che ruotava di più o di meno, per questo hanno invertito le risposte. Dal grafico si nota che nel secondo test dopo le lezioni con Logo ci sono state quattro risposte corrette in più rispetto ad una sola dopo l'utilizzo di Scratch.

Domanda 6

Test 1. Un robot ruota di 90 gradi ogni volta che ruota. Quante rotazioni deve fare il robot per puntare nella stessa direzione? Perché?

Test 2. Un robot ruota di 60 gradi ogni volta. Quante rotazioni deve fare prima di tornane nella stessa posizione iniziale? Perché?

Test 3. Un robot ruota di 120 gradi ogni volta che ruota. Quante rotazioni deve fare il robot per puntare nella stessa direzione? Perché?



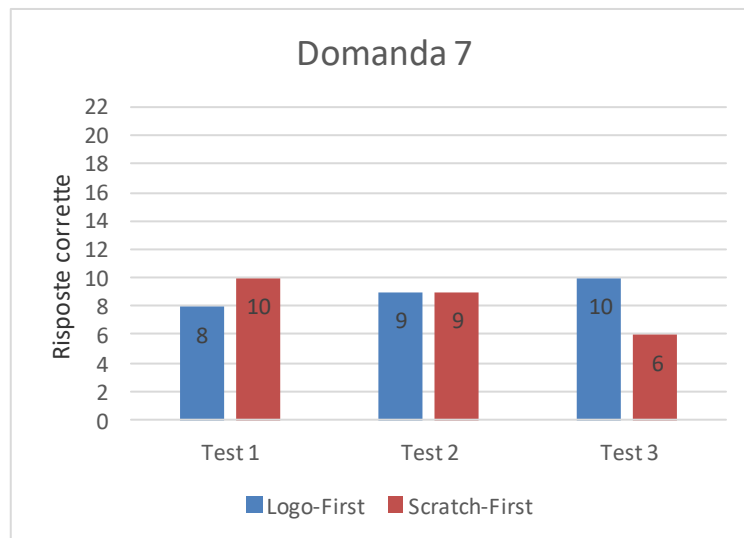
Osservando il grafico possiamo notare che ci sono state delle variazioni anche se lievi specialmente per il gruppo *Logo-First*. Nel secondo test però il gruppo *Scratch-First* ha peggiorato i suoi risultati, anche se di un solo bambino. Abbiamo però notato che molti dei bambini che hanno risposto in modo errato avevano letto male il testo dando per scontato che la risposta fosse uguale a quella del test precedente poiché la domanda era la stessa ma cambiavano i valori di rotazione, hanno quindi dato una risposta corretta ma non rispetto alla domanda (ad esempio hanno scritto “*il robot deve fare 4 rotazioni, perché $90 \times 4 = 360$* ”).

Domanda 7

Test 1. Un robot ruota di 30 gradi ogni volta. Quante rotazioni deve fare prima di puntare la stessa direzione? Perché?

Test 2. Un robot ruota di 90 gradi ogni volta che ruota. Quante rotazioni deve fare il robot per puntare nella direzione opposta? Perché?

Test 3. Un robot ruota di 45 gradi ogni volta che ruota. Quante rotazioni deve fare il robot per puntare nella direzione opposta?



Anche in questo caso le variazioni non sono elevate, per quanto riguarda il gruppo *Scratch-First* sembra che siano peggiorati, specialmente nell'ultimo test, ma analizzando le risposte spesso, come per la domanda precedente, non era stata letta bene la richiesta e quindi invece di indicare il risultato per far puntare il robot nella direzione opposta hanno indicato come farlo tornare nella direzione iniziale.

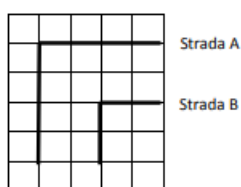
Domanda 8

Test 1: Nella città chiamata Quadrata tutte le strade sono disposte su uno schema quadrato. Ci sono solo due strade principali, che sulla mappa sono rappresentate da linee più scure.

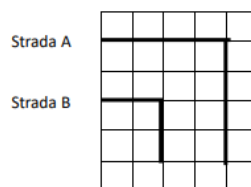
Seleziona la risposta che pensi sia corretta:

- La strada A curva di più della strada B
- La strada B curva di più della strada A
- La strada A e la strada B curvano nello stesso modo
- Non puoi dirlo

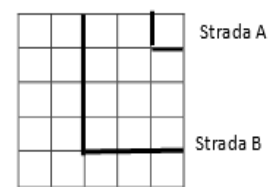
Test 1

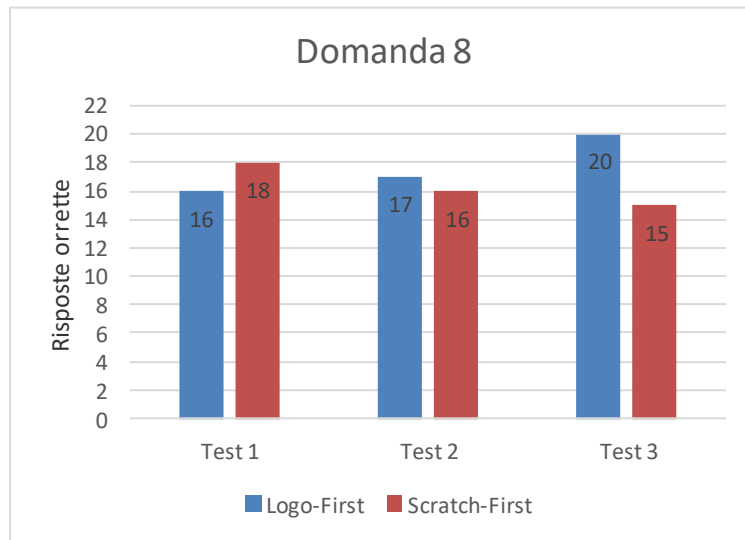


Test 2



Test 3





Dai grafici possiamo notare che la maggior parte dei bambini ha compreso che essendo angoli retti ruotano nello stesso modo. Alcuni dei bambini hanno proprio scritto il valore dei due angoli sul test e hanno poi risposto correttamente. Da notare come se per il gruppo *Logo-First* i risultati sono migliorati nei tre test, nel gruppo *Scratch-First* sono invece progressivamente peggiorati.

Domanda 9

Nella città di Quadrata ci sono due nuove strade C e D

Seleziona la risposta che pensi sia corretta:

La strada C curva più della strada D

La strada D curva più della strada C

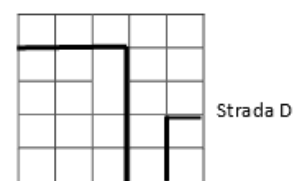
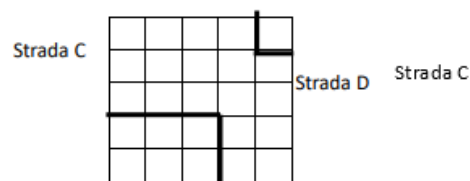
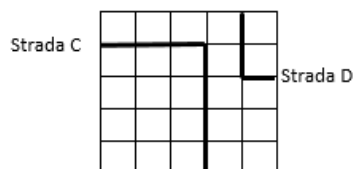
Le strade C e D curvano nello stesso modo

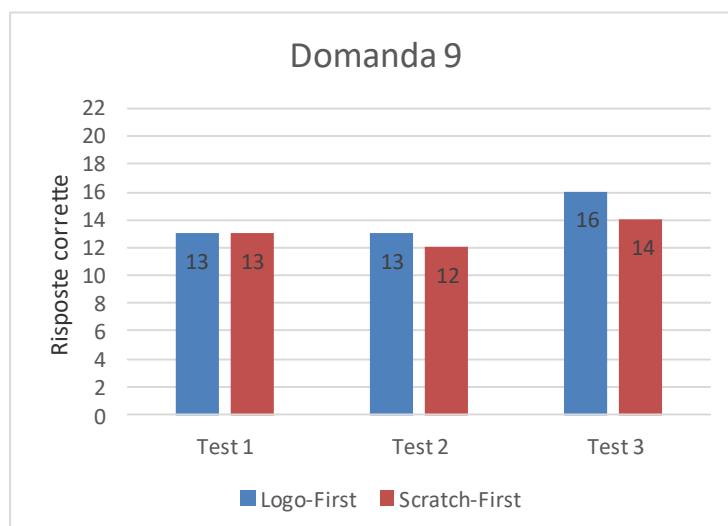
Non puoi dirlo

Test 1

Test 2

Test 3





I risultati dei primi due test non mostrano cambiamenti significativi, ma nell'ultimo test le risposte sono migliorate specialmente per il gruppo *Logo-First*. Risulta interessante notare che le risposte corrette sono notevolmente diminuite rispetto alla domanda precedente che presentava la stessa domanda, ma nella quali i due angoli sono posizionati nello stesso modo. Cambiando la posizione dei due angoli gli errori sono aumentati.

Domanda 10

Seleziona la risposta che pensi sia corretta:

L'angolo A è maggiore dell'angolo B

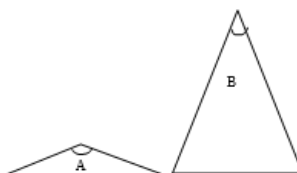
L'angolo B è maggiore dell'angolo A

Gli angoli A e B hanno la stessa ampiezza

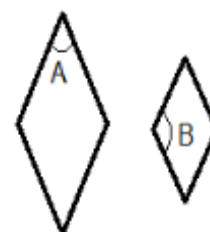
Non puoi dirlo



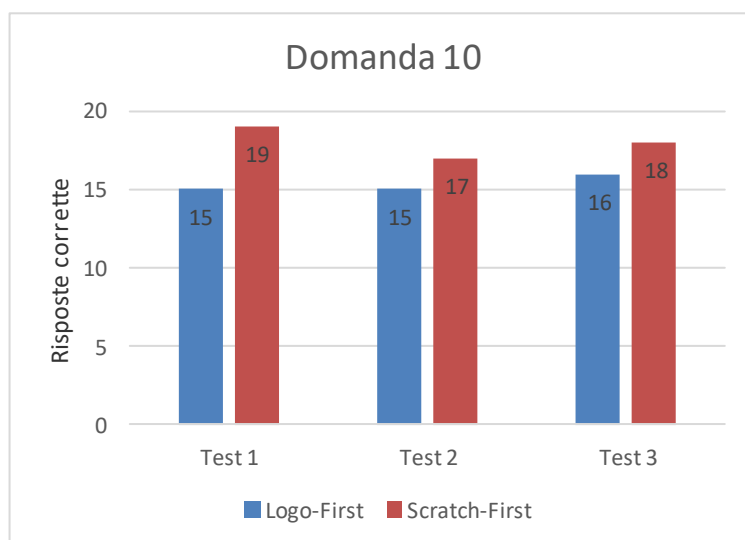
Test 1



Test 2



Test 3



Nonostante non ci siano state grandi variazioni risulta significativo che alcuni bambini abbiano dato una risposta scorretta poiché probabilmente ancora collegano l'ampiezza dell'angolo alla lunghezza dei lati della figura.

Domanda 11

In ognuno di questi triangoli tutti i lati hanno la stessa lunghezza.

Seleziona la risposta che pensi sia corretta:

L'angolo C è maggiore dell'angolo D

L'angolo D è maggiore dell'angolo C

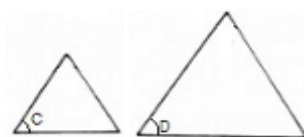
Gli angoli C e D hanno la stessa ampiezza

Non puoi dirlo

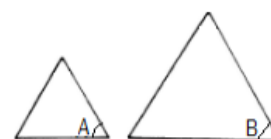
Test 1

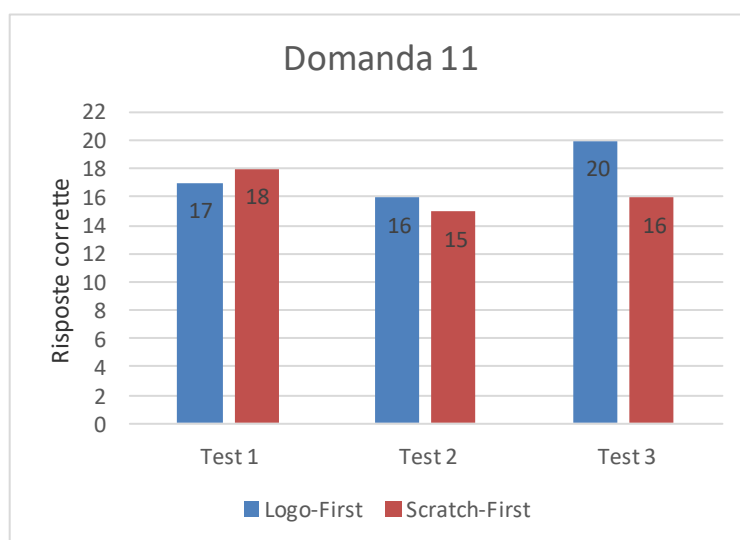


Test 2



Test 3

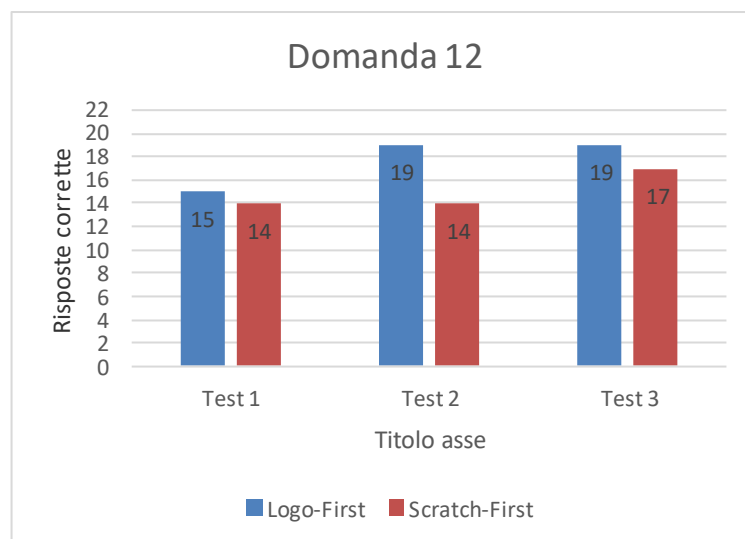
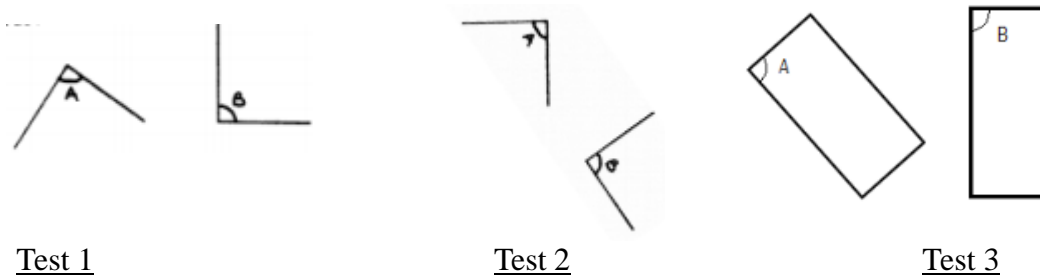




Dai risultati notiamo che alcuni bambini non hanno risposto correttamente e, quindi, o non hanno compreso che il triangolo equilatero possiede tutti gli angoli uguali, anche se quando costruivano i triangoli equilateri usavano la stessa ampiezza per tutti i lati, o non hanno collegato la definizione del testo ad un triangolo equilatero e la figura che non rappresenta i soliti segni che nei loro libri o nelle verifiche rappresentano il triangolo equilatero (i trattini sui lati o i tre cerchietti uguali sugli angoli) non è quindi stata riconosciuta.

Domanda 12

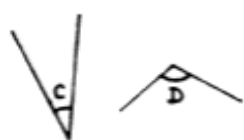
- L'angolo A è maggiore dell'angolo B
- L'angolo B è maggiore dell'angolo A
- Gli angoli A e B hanno la stessa ampiezza
- Non puoi dirlo



I risultati mostrano che la maggioranza dei bambini ha risposto correttamente, ma alcuni ancora hanno difficoltà a riconoscere l'angolo non posizionato nel modo convenzionale. Inoltre è possibile notare dal secondo test che chi ha utilizzato Logo è migliorato, mentre, chi ha utilizzato Scratch non ha mostrato miglioramenti, i dati del Test 3, dopo che i gruppi si sono invertiti, mostrano ancora una volta che chi ha usato Scratch non ha mostrato miglioramenti mentre chi ha utilizzato Logo è migliorato.

Domanda 13

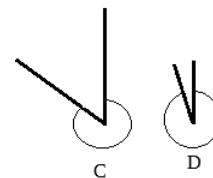
- L'angolo C è maggiore dell'angolo D
- L'angolo D è maggiore dell'angolo C
- Gli angoli C e D hanno la stessa ampiezza
- Non puoi dirlo



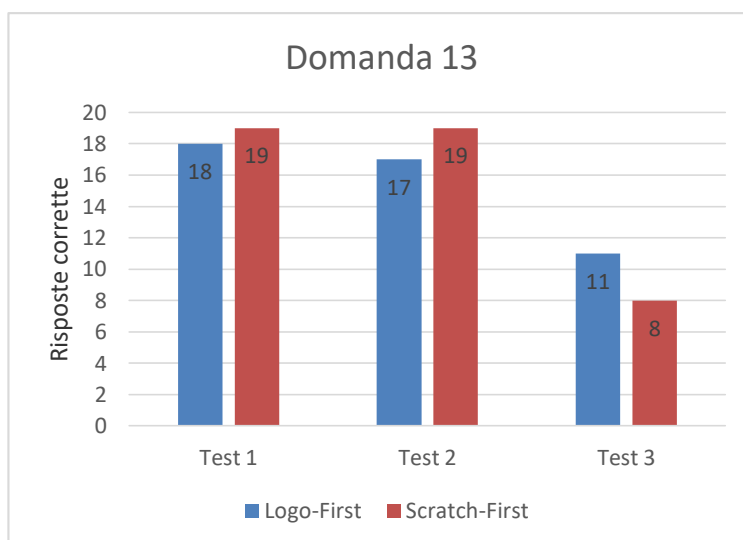
Test 1



Test 2



Test 3



Dal grafico notiamo che tra il primo e il secondo test non ci sono state notevoli variazioni e la maggioranza dei bambini ha risposto correttamente. Nel terzo test invece i risultati sono peggiorati notevolmente, ma è necessario tenere in considerazione che gli angoli proposti si differenziavano dai primi due test poiché veniva richiesto l'angolo esterno e molti bambini potrebbero aver sbagliato poiché hanno considerato l'altro angolo.

Per confrontare al meglio i risultati del Test 1 e del Test 2 che permettono di vedere la variazione tra l'utilizzo di Logo e di Scratch, abbiamo costruito la seguente tabella.

N° Domanda	Gruppo Logo-First			Gruppo Scratch-First		
	N° Risposte corrette Test 1	N° Risposte corrette Test 2	Variazione Risposte corrette	N° Risposte corrette Test 1	N° Risposte corrette Test 2	Variazione Risposte corrette
1	2	2	0	2	4	+2
2	11	18	+7	11	13	+2
3	22	20	-2	20	19	-1
4	21	20	-1	22	21	-1
5	2	6	+4	6	7	+1
6	10	12	+2	11	10	-1
7	8	9	-1	10	9	-1
8	16	17	+1	18	16	-2
9	13	13	0	13	12	-1
10	15	15	0	19	17	-2
11	17	16	-1	18	15	-3
12	15	19	+4	14	14	0
13	18	17	-1	19	19	0

Ci siamo concentrati sull'osservazione della variazione delle risposte corrette a seconda del linguaggio di programmazione utilizzato. Il gruppo *Logo-First* che ha utilizzato Logo ha mostrato, rispetto all'altro gruppo, un maggiore numero di risposte corrette, infatti, ci sono state diciotto risposte corrette in più rispetto al primo test, e anche un minor numero di risposte errate in più, le risposte corrette sono diminuite solamente in sei casi. Il gruppo *Scratch-First*, invece, ha mostrato solamente cinque risposte corrette in più e dodici risposte corrette in meno, rispetto al primo test.

3.8 Presentazione del questionario

Alla fine del percorso volevamo anche raccogliere dati per capire come i bambini avevano vissuto l'esperienza e analizzare in generale come solitamente vivono e utilizzano le tecnologie. In totale il questionario presenta diciotto domande, alcune sono domande chiuse e possono quindi prevedere più di una risposta (per questo nei grafici può risultare maggiore il numero delle risposte rispetto al totale dei bambini) e alcune domande chiuse. È possibile dividere le tre domande in tre aree distinte:

1. Domande legate all'esperienza con Logo e Scratch. I bambini hanno potuto esprimere un parere su questa esperienza e segnalare come hanno vissuto le attività con Logo e con Scratch, indicandone aspetti positivi ed eventuali difficoltà riscontrate, hanno poi segnalato un'eventuale preferenza tra i due linguaggi di programmazione, motivando questa scelta. Poi è stato domandato se secondo loro l'utilizzo di questi programmi potesse insegnare qualcosa. Infine abbiamo chiesto se conoscevano già i programmi e se avrebbero voluto continuare a usarli.

2. Domande legate al loro utilizzo delle tecnologie. È stato chiesto ai bambini se a casa utilizzavano le tecnologie specificando quali e per quanto tempo. Anche in questo caso abbiamo domandato se ritenevano che le tecnologie che utilizzano a casa insegnino loro qualcosa e, in caso affermativo, che cosa.

3. Domanda relative al lavoro di coppia. Abbiamo chiesto ai bambini come avessero vissuto il lavoro di coppia, concentrandoci su i vantaggi e le difficoltà riscontrati.

Il tempo a disposizione per poter svolgere il questionario era di trenta minuti e tutti i bambini delle due classi hanno potuto rispondere alle domande.

Riporto in seguito le domande del questionario somministrato ai bambini.

Questionario

Quale programma hai utilizzato per primo?

- Logo
- Scratch

1. Quanto ti è piaciuta, in generale, questa esperienza?

- Per nulla
- Poco
- Abbastanza
- Molto
- Moltissimo

2. Come ti sono sembrate le lezioni con Logo?

- Stimolanti
- Noiose
- Divertenti
- Interessanti
- Alternative
- Altro.....

3. Come ti sono sembrate le lezioni con Scratch?

- Stimolanti
- Noiose
- Divertenti
- Interessanti
- Alternative
- Altro.....

4. Con quale programma ti è piaciuto di più lavorare?

- Scratch

Logo

5. Perché?

.....
.....
.....

6. Quali difficoltà hai avuto con Logo?

.....
.....
.....

7. Quali difficoltà hai avuto con Scratch?

.....
.....
.....

8. Quando hai giocato con Logo, pensi di aver imparato qualcosa? (Puoi scegliere più di un'opzione)

- No, ho solo giocato
- No, mi sono annoiato
- Sì, ma non ho capito molto
- Sì, ho imparato a comprendere meglio gli angoli
- Sì, ho imparato che la tartaruga si muove solo con delle indicazioni ben precise
- Sì, ho imparato a programmare la tartaruga in modo da farla disegnare
- Sì, ho imparato ad utilizzare meglio il computer
- Altro.....

9. Quando hai giocato con Scratch, pensi di aver imparato qualcosa? (Puoi scegliere più di un'opzione)

- No, ho solo giocato
- No, mi sono annoiato
- Sì, ma non ho capito molto
- Sì, ho imparato a comprendere meglio gli angoli
- Sì, ho imparato che il gatto si muove solo con delle indicazioni ben precise
- Sì, ho imparato a programmare il gatto in modo da farlo disegnare
- Sì, ho imparato ad utilizzare meglio il computer
- Altro.....

10. Hai mai utilizzato questi programmi prima di queste lezioni?

- No
- Sì, ho utilizzato Scratch
- Sì, ho utilizzato Logo
- Sì, ho utilizzato entrambi

11. Ti piacerebbe utilizzare anche a casa questi programmi?

- No
- Sì, mi piacerebbe utilizzare Logo
- Sì, mi piacerebbe utilizzare Scratch
- Sì, mi piacerebbe utilizzare entrambi

12. A casa utilizzi le tecnologie?

- Per nulla
- Poco
- Abbastanza
- Molto
- Moltissimo

13. Quale tecnologia utilizzi di più?

- TV
- Computer
- Videogames
- Cellulare
- Tablet
- Altro.....

14. Per quanto tempo le utilizzi durante la tua giornata?

- Mai
- Poco (10/20 minuti)
- Abbastanza (mezz'ora/ un'ora)
- Molto (un'ora/due ore)
- Moltissimo (più di due ore)

15. Credi che ti aiutino ad imparare qualcosa?

- Sì
- No

16. Se sì, che cosa?

.....

.....

.....

Ti è piaciuto lavorare in coppia con il tuo compagno?

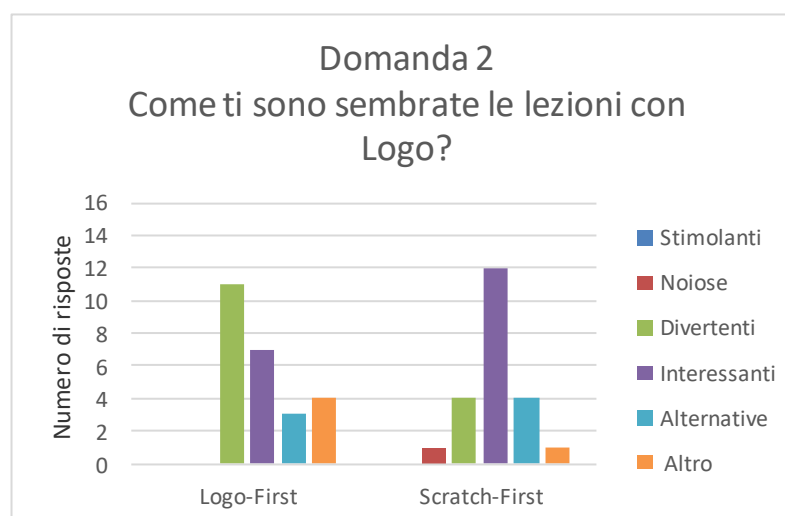
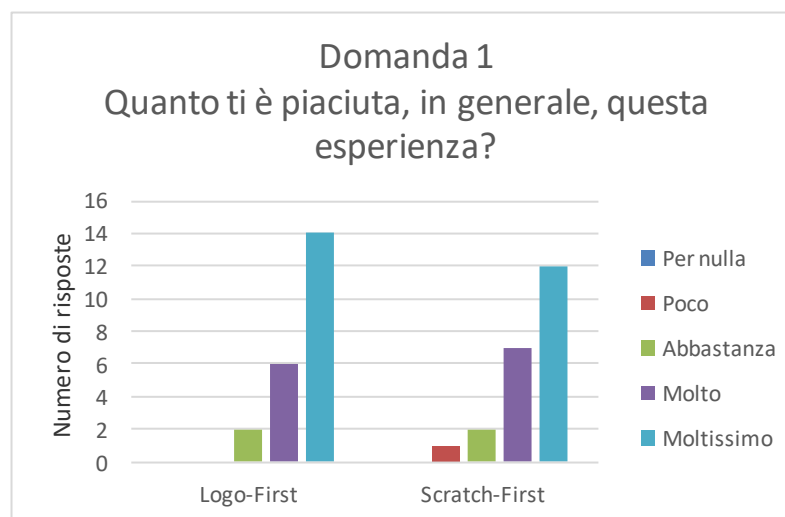
- Per nulla
- Poco
- Abbastanza
- Molto
- Moltissimo

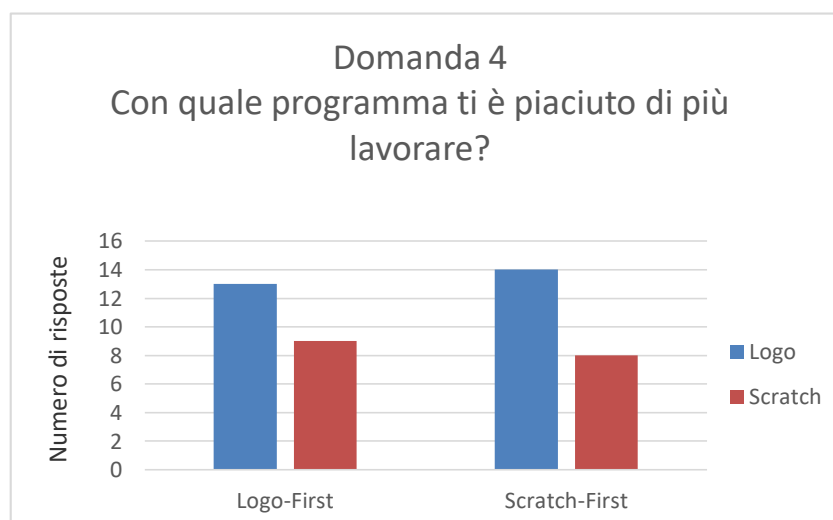
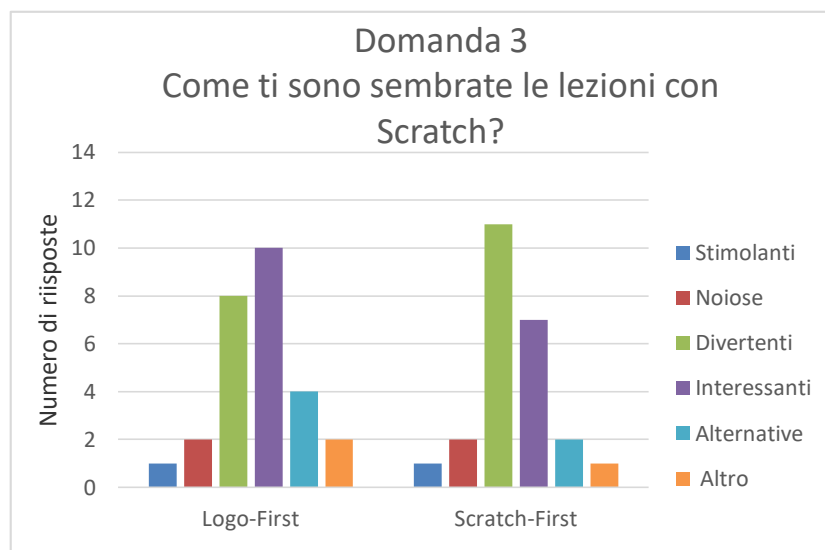
17. Nel lavoro di coppia:

- Il compagno mi ha aiutato a capire meglio ciò che dovevamo fare
- Il mio compagno mi ha ostacolato nel lavoro
- Il mio compagno non mi ha permesso di utilizzare il programma
- Siamo riusciti a lavorare insieme

3.9 Analisi dei risultati del questionario

Per analizzare questi dati abbiamo deciso di continuare la divisione tra i due gruppi in modo da poter riflettere, anche per alcune di queste domande, su se e come i due programmi possano aver creato delle differenze nei risultati dei bambini. Abbiamo deciso di utilizzare nuovamente un istogramma che permette di confrontare in modo chiaro i risultati.





Domanda 5. Perché?

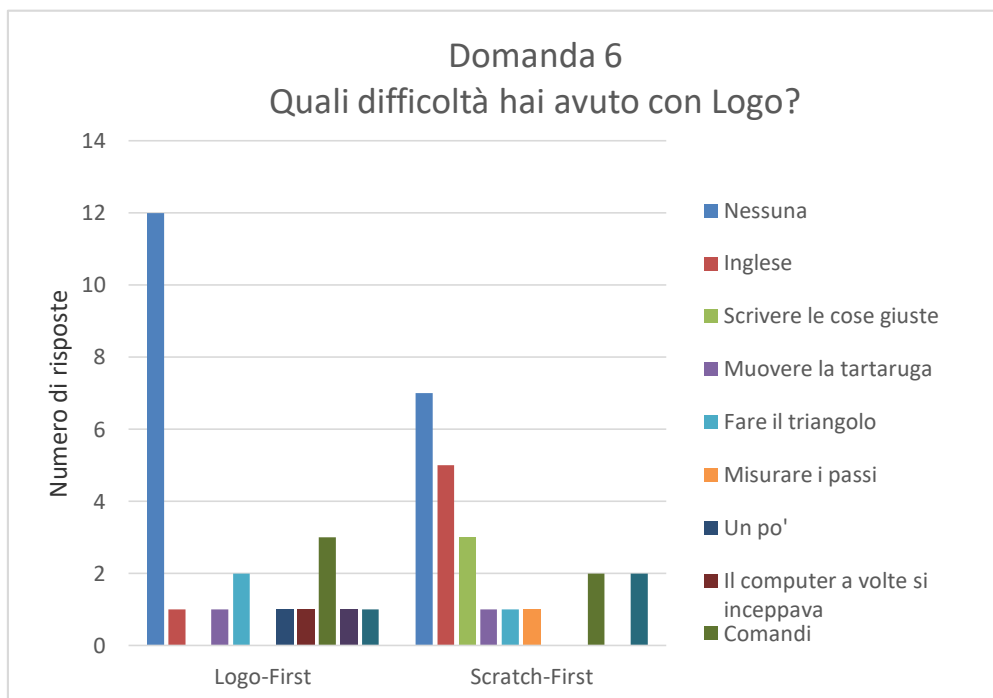
I bambini che hanno preferito Logo hanno motivato la scelta con le seguenti risposte:

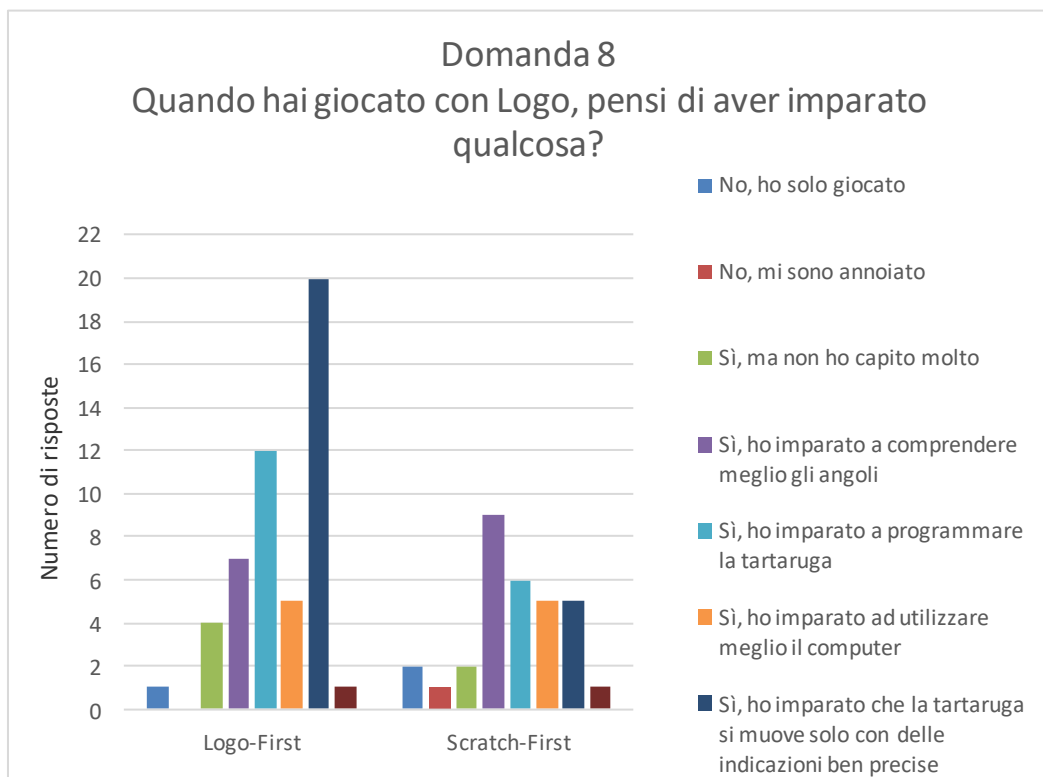
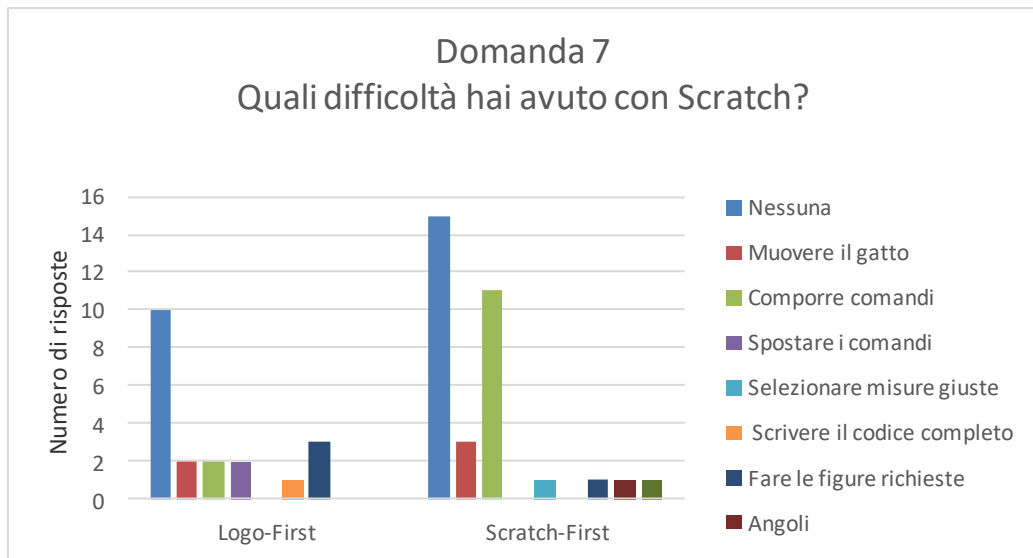
È più divertente; Insegna l'inglese; Insegna nuove parole inglesi; È più interessante; Era un'esperienza nuova; Mi trovavo meglio con i comandi; Comandi più precisi e più colori; Scratch carino ma Logo

invincibile; Puoi creare più cose ed è più divertente; Dovevi scrivere tu i comandi e più creativo; Più bello, scrivi tu i comandi e c'è la tartaruga; Scrivere i comandi quindi più bello; Tante cose da scoprire, se sbagli vengono cose carine; Divertente mentre Scratch non ho fatto i giochi; Devi ragionare di più e quindi più divertente; Più semplificato non bisogna prendere i comandi; Dovevi scrivere tu i comandi.

I bambini che hanno preferito Scratch hanno motivato la scelta con le seguenti risposte:

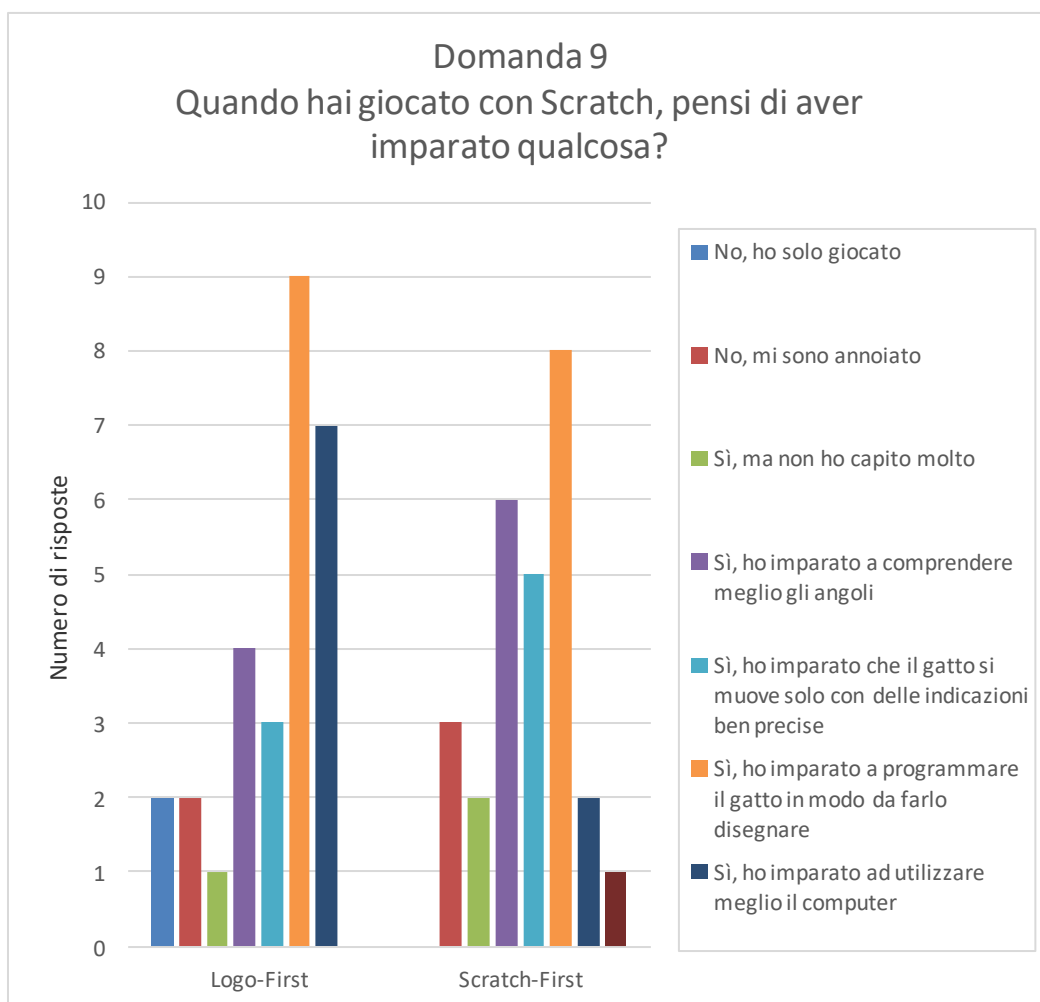
È più divertente; È più colorato; Sono abituato ai comandi; È un'abitudine; C'erano più scelte per farlo muovere; Più complicato dovevamo ragionare di più; Più opzioni e i comandi non erano in inglese; Potevo fare più cose strane; Mi piace il gioco in sé.





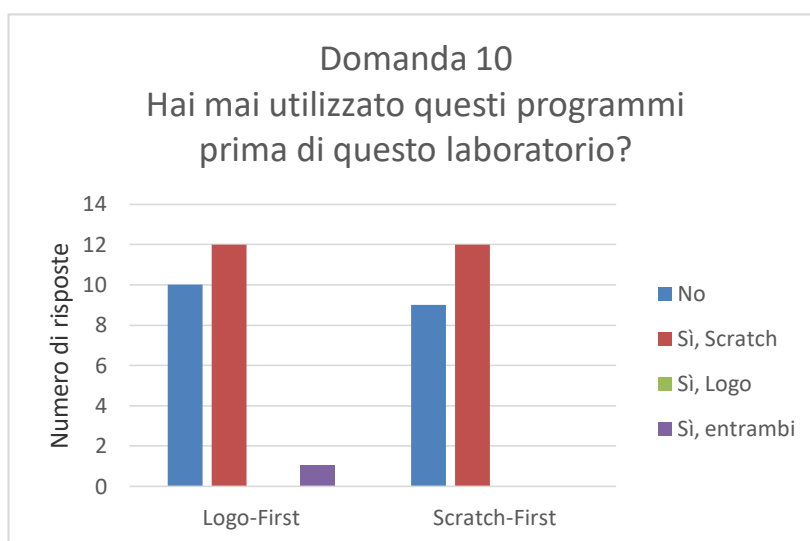
I bambini che hanno risposto a questa domanda con “Altro”, hanno aggiunto:

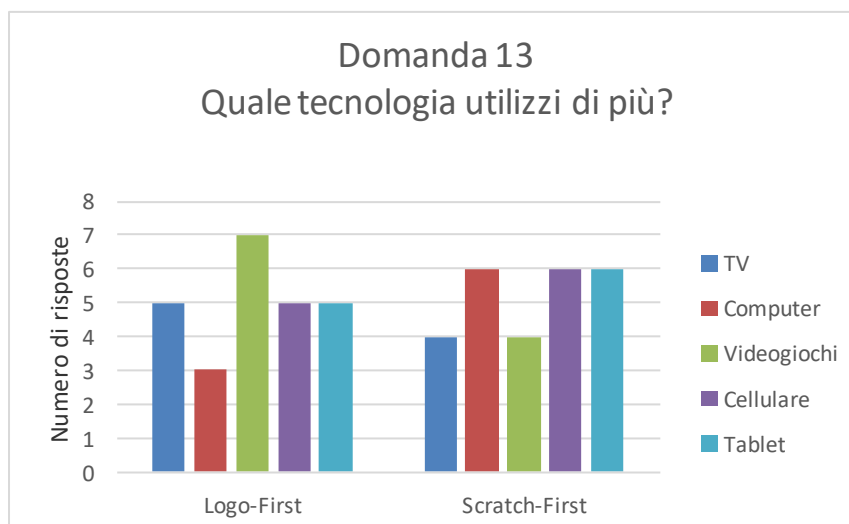
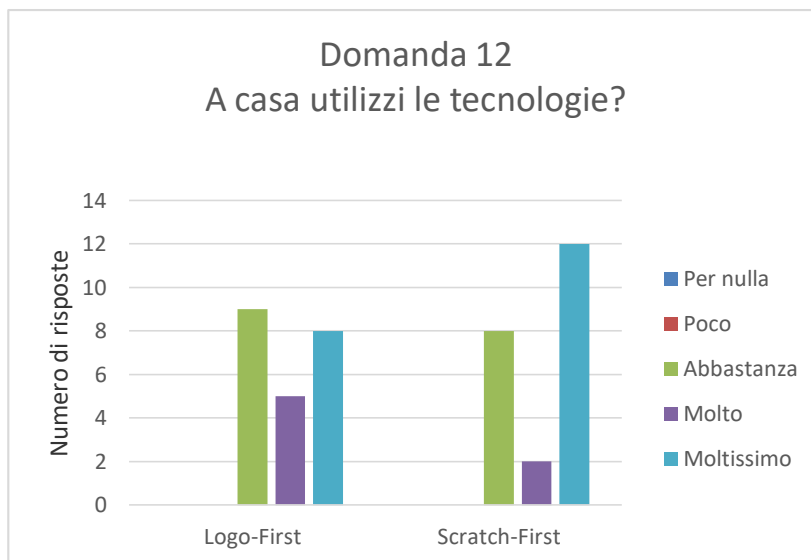
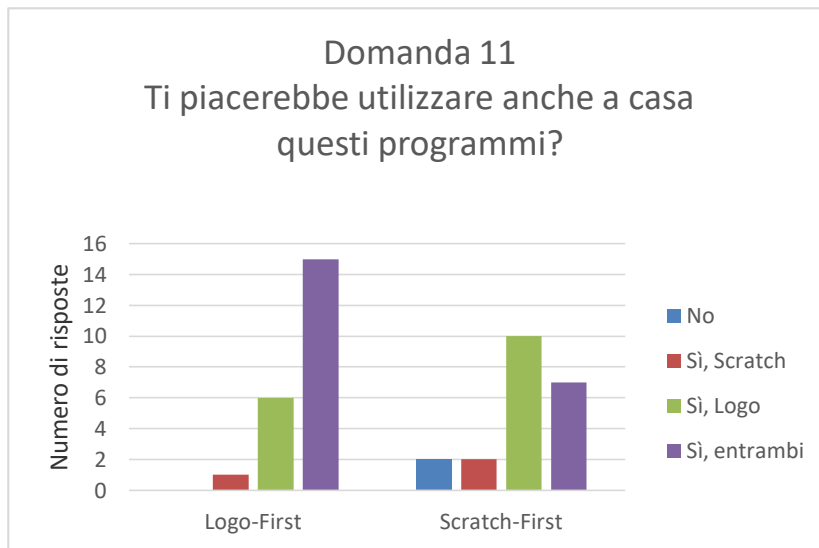
Ho imparato l'inglese; Ho imparato a ragionare meglio; Ho

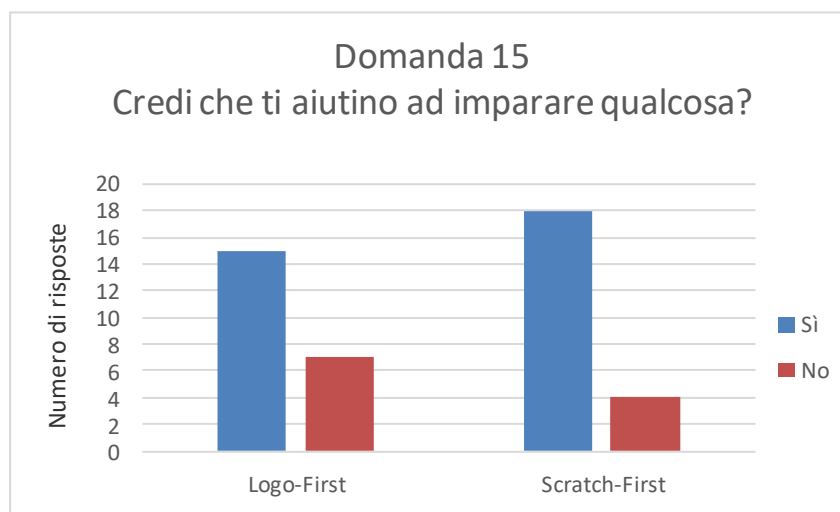
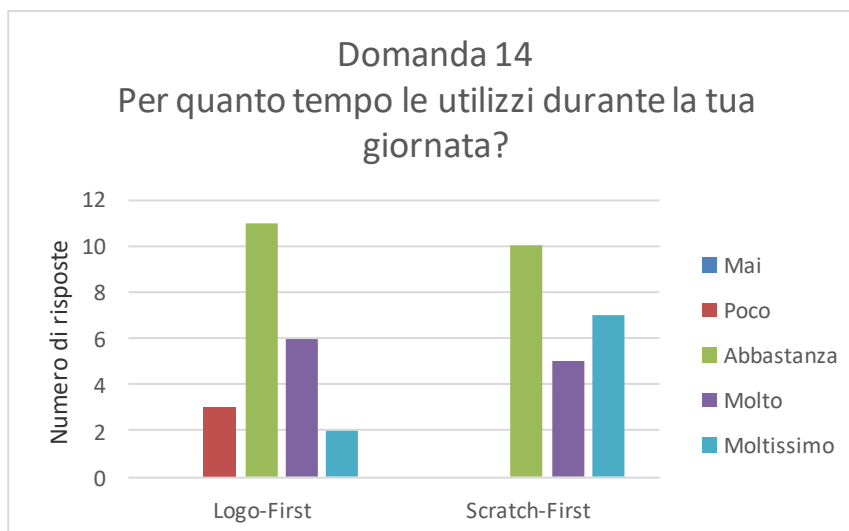


imparato a disegnare e colorare cose complesse.

Il bambino che ha risposto “Altro” a questa domanda, ha aggiunto che con Scratch “ho imparato a ragionare meglio”.





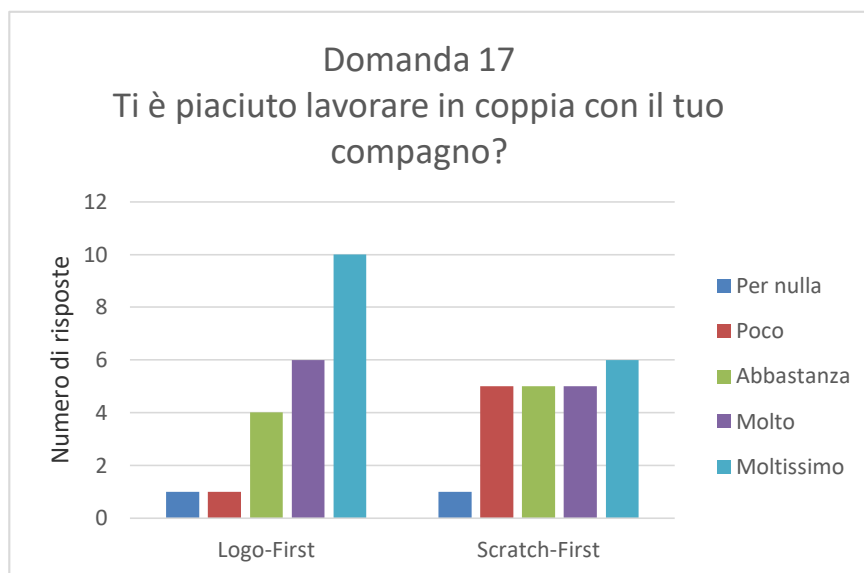


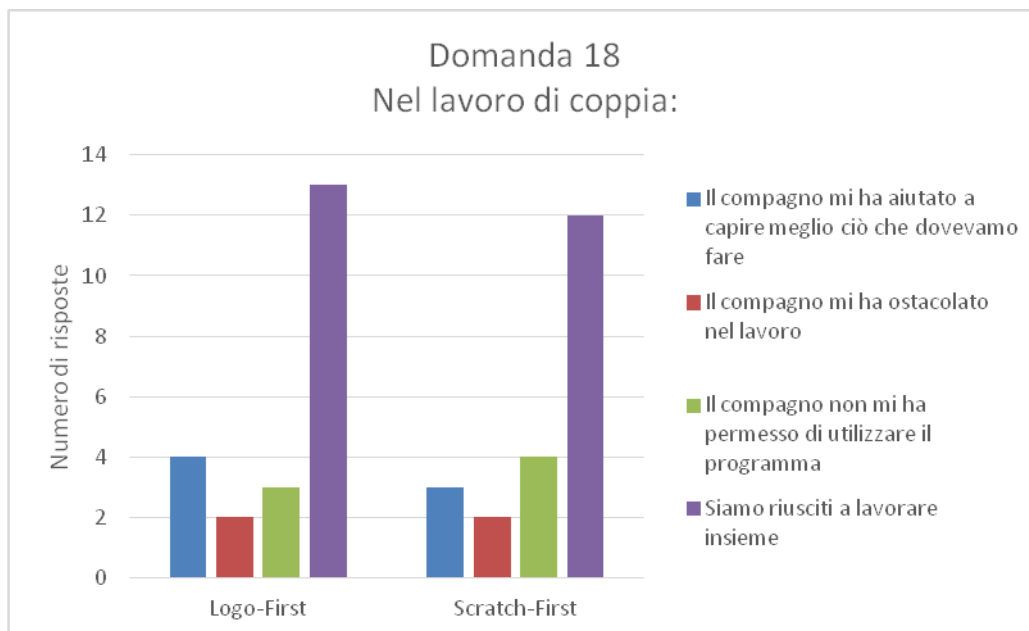
Domanda 16. Se sì, che cosa?

Riporto qui le testuali parole dei bambini:

rilassarmi; sviluppare l'intelligenza; essere educato; imparare a parlare ben; avere più esperienza per fare le cose; imparare ad utilizzare nuove cose; avere più esperienza; i comandi; imparare le lingue e fare gli esercizi online; documentari; come utilizzare le cose

che hai; creatività; che senza tecnologia non si vive; cose di geometrie e altre cose; a migliorare il modo di giocare a calcio; ogni curiosità puoi cercarla su Google; ad interagire con il mondo virtuale; certe parole; notizie che ti tengono aggiornato; navigare, evitare pubblicità ordinare alberghi e molto altro; essere attenta nelle ricerche e quindi a quello che scrivo; il linguaggio, la scrittura e le doppie; per fare nuove esperienze con programmi nuovi; a imparare gli angoli; guardo video su esperimenti e poi li faccio.





Dalle prime domande è possibile osservare che i bambini hanno vissuto l'esperienza in modo positivo; sia Logo che Scratch sono stati ritenuti per la maggioranza dei bambini interessanti e divertenti, ma è un dato da osservare che, se per quanto riguarda Scratch i dati tra i due gruppi non mostrano differenze significative, nella domanda su Logo il gruppo *Logo-First* le ha ritenute principalmente divertenti e nessuno le ha ritenute noiose, mentre il gruppo *Scratch-First* le ha ritenute principalmente interessanti.

Dai dati emerge poi che il programma preferito da entrambi i gruppi è stato Logo, interessanti sono state le motivazioni di questa scelta, infatti, molti bambini hanno indicato proprio il dover scrivere il codice come punto di forza del programma, ma è stato scelto anche perché viene ritenuto più motivante potendo fare più cose e dovendo scrivere in inglese potevano imparare di più. Molto interessante è stata una risposta che ha valorizzato l'errore dicendo "*ci sono tante cose da scoprire, se sbagli vengono cose carine*" quindi questo bambino ha vissuto l'errore come auspicato da Papert come un momento di scoperta e apprendimento e non come qualcosa di negativo da eliminare. Chi ha preferito Scratch principalmente ha dato valore all'aspetto del software più colorato e che offriva più scelte, alcuni hanno preferito Scratch poiché i blocchi sono percepiti come più semplici e sono in

italiano, una risposta interessante che si distacca dalle altre è stata “*Più complicato dovevamo ragionare di più*”.

Analizzando invece le difficoltà riscontrate con Logo la maggioranza dei bambini non ha evidenziato particolari difficoltà, anche se sono state riscontrate maggiori difficoltà dal gruppo *Scratch-First*, che ha indicato come principale problematicità l'utilizzo dell'inglese. L'altra difficoltà che accomuna i due gruppi sono i comandi ma il gruppo *Scratch-First* ha anche riscontrato molte difficoltà nello scrivere la cosa giusta, è possibile che quindi usando prima un linguaggio a blocchi, successivamente si percepiscano maggiori le difficoltà legate agli errori sintattici. Interessanti anche i dati relativi alle difficoltà con Scratch, anche in questo caso la maggioranza dei bambini non ha indicato difficoltà, ma è stata riscontrata una differenza tra i due gruppi, il gruppo *Logo-First* ha mostrato maggiore difficoltà rispetto al gruppo *Scratch-First*. Da questi dati emerge che il programma utilizzato per primo tende a risultare più semplice mentre cambiando tipologia di linguaggio aumentano le difficoltà. Nonostante questi dati è anche interessante notare la differenza tra le difficoltà riscontrate, nel gruppo *Logo-First* le difficoltà sono state di diversa tipologia distribuendosi equamente tra le possibili risposte, mentre il gruppo *Scratch-First* ha individuato come difficoltà maggiore la composizione dei comandi.

Per quanto riguarda cosa i bambini hanno appreso durante l'utilizzo di questi due linguaggi di programmazione, in entrambi i casi è stato indicato di aver imparato a programmare, di aver migliorate le loro conoscenze sugli angoli e di aver capito che servono indicazioni precise nella programmazione. Interessanti le risposte libere che potevano mettere perché per Logo nuovamente l'inglese viene individuato come punto di forza, ma soprattutto, per entrambi i programmi è stato scritto che hanno imparato a ragionare meglio.

Dai dati è anche emerso che diversi bambini avevano già utilizzato Scratch, quindi risulta interessante come nonostante questo, Logo rimanga il preferito e anche quando dovevano indicare se volevano continuare ad utilizzare il programma in molti hanno indicato solo Logo, diversi però hanno anche indicato entrambi.

Riguardo all'utilizzo delle tecnologie a casa è emerso che la maggioranza dei bambini le utilizza "Abbastanza" "Molto" e "Moltissimo", per quanto riguarda la tipologia utilizzano molte tecnologie diverse e per diversi bambini insegnano loro qualcosa.

Infine i dati sul lavoro di gruppo mostrano che nonostante la maggioranza abbia indicato "Siamo riusciti a lavorare insieme" per alcuni è stato difficile lavorare in coppia spesso perché il compagno non gli permetteva di utilizzare il computer.

Conclusioni

Tramite questa esperienza abbiamo potuto raccogliere dati interessanti sull'utilizzo di questi due linguaggi di programmazione. Sono consapevole che, nonostante il campione sia stato ampliato grazie alla possibilità di svolgere la tesi in collaborazione con Martina Fini, rimanga un piccolo campione e che il tempo a disposizione non sia stato molto, ma ritengo comunque possibile fare riflessioni interessanti a partire dai dati raccolti e magari potrebbe essere una piccola base per delle ricerche future. Sicuramente potrebbe essere un'esperienza didattica da svolgere in futuro con le mie classi, i tempi richiesti, che per una ricerca possono essere pochi, in realtà per una classe sono impegnativi, ma è possibile in un intero anno svolgere un buon corso di coding, naturalmente è anche necessario avere delle attrezzature funzionanti e per quanto riguarda Scratch una buona connessione, ma con un'attenta preparazione è possibile organizzare queste attività anche con strumenti diversi (ad esempio è possibile progettare anche lezioni di *coding unplugged*). Poter osservare i bambini durante l'utilizzo di questo programma mi ha portato a riflettere su quello che avevo letto in teoria sul coding e sulla sua valenza educativa. Soprattutto per quanto riguarda la visione dell'errore, nella prima lezione un bambino mi ha chiesto preoccupato tra le prime domande "*Cosa succede se sbagliamo?*" o durante le prime prove mi chiamavano preoccupati quando succedeva qualcosa che non avevano preventivato, ma soprattutto durante l'utilizzo di Logo, quando hanno iniziato a creare delle figure interessanti e probabilmente anche vedendo come reagivo davanti all'errore, hanno iniziato a cambiare la loro visione e infatti, come ho già scritto, un bambino ha preferito Logo proprio perché dagli errori scoprivano cose nuove e per loro belle. Interessante è stato osservare le diverse strategie utilizzate per creare le figure richieste, alcuni provavano direttamente scrivendo il codice e vedendo cosa accadeva mentre altri avevano bisogno di fogli o hanno chiesto di utilizzare la lavagna per poter prima progettare cosa avrebbero poi provato a far fare al programma; ho collegato questi stili a quelli che Papert definisce dei *planners* e dei *bricoulers* e come questi ultimi tendenzialmente abbiano lavorato con meno difficoltà e spesso con più entusiasmo.

Il lavoro in coppia ha svolto un ruolo fondamentale, nonostante per alcuni sia stata una difficoltà maggiore perché il compagno voleva lavorare da solo e non condivideva il computer. Per la maggioranza dei bambini ha avuto invece un ruolo positivo, specialmente le due bambine DSA hanno tratto beneficio dal lavoro con un compagno che non solo le ha aiutate nello svolgimento dei compiti richiesti, ma ho visto che loro stesse sono state attive e hanno fatto proposte durante il lavoro, una delle due bambine nel questionario in relazione alle domande sulle difficoltà ha scritto che non ne ha avute proprio perché ha lavorato con la sua compagna. Inoltre, credevo che avrebbero avuto più difficoltà con Logo invece entrambe lo hanno preferito e durante le attività una delle bambine mi ha detto che era molto contenta perché le piaceva riuscire a creare delle cose e questo mi ha dato molta soddisfazione. Un altro bambino mi ha piacevolmente sorpreso, solitamente non partecipa molto alle attività, anche per la sua timidezza, invece, durante le lezioni si è spesso messo in gioco venendo anche a spiegare alla lavagna agli altri compagni il ragionamento che aveva fatto, durante una delle lezioni mancava il suo compagno allora lui ha deciso, nel momento di libera creazione, di disegnare qualcosa di nuovo per il compagno assente per poi spiegargli come aveva fatto.

Riprendendo gli obiettivi stabiliti inizialmente possiamo notare dall'analisi dei dati che per quanto riguarda l'utilizzo di Logo gli si possa attribuire un valore positivo per l'apprendimento di alcune conoscenze legate al concetto di angolo, mentre con Scratch sono state osservate correlazioni negative. Come già detto i fattori che hanno portato a questi dati potrebbero essere diversi, ma per quanto riguarda questa ricerca possiamo affermare che il linguaggio di programmazione testuale risulta maggiormente efficace rispetto ad un linguaggio visuale nell'apprendimento di concetti geometrici specialmente riguardanti gli angoli. Abbiamo anche avuto la possibilità, finite le lezioni, di analizzare con l'insegnante di geometria alcuni dei punti che ancora potevano risultare ostici per alcuni bambini ed è stato interessante osservare insieme quello che avevano capito e i punti dove ancora potevamo lavorare.

I bambini hanno anche appreso alcuni costrutti base della programmazione ma soprattutto per alcuni di loro potrebbe essere nato un interesse verso il coding,

molti finite le attività mi hanno detto che hanno immediatamente installato Logo e hanno provato a creare cose nuove, mi hanno chiesto se esistevano dei codici per potergli far fare alcune cose che avevano in mente e questo mi ha dato tanta soddisfazione.

Tenendo in considerazione questi aspetti ritengo questa esperienza positiva: per me è stato un'occasione per mettermi in gioco come insegnante e per approfondire le mie competenze con questi linguaggi e testare il loro valore educativo, ma anche per i bambini è stata un'esperienza che hanno vissuto positivamente e che ha permesso loro di apprendere qualcosa di nuovo che potrà essere utile sia nel loro percorso formativo che nella loro vita extrascolastica.

Bibliografia

- Abelson, H., & diSessa, A. (1986). *Turtle Geometry*. Cambridge, Massachusetts: The MIT Press.
- Avalle, U., & Maranzana, M. (2005). *Pensare ed educare*. Cuneo: Paravia.
- Barbero, A. (2014, Giugno). La programmazione "from Scratch". *Bricks*, 4(2), 34-42.
- Barbero, A. (2014, Novembre). Programmare con Scratch. *Science Magazine*(1), 17-19.
- Bennett, S. J., & Maton, K. (2010). Beyond the 'digital natives' debate: towards a more nuanced understanding of students' technology. *Journal of Computer Assisted Learning*, 26(5), 321-331.
- Bennett, S., Maton, K., & L., K. (2008). The 'Digital Natives' Debate: A Critical Review of the Evidence. *British Journal of Educational Technology*, 39(5), 775–786.
- Bogliolo, A. (2016). *Coding in Your Classroom, Now!* Firenze: Giunti.
- Brennan, K., & Resnick, M. (2012). *New Frameworks for Studying and Assessing the Development of Computational Thinking*. Proceeding of the 2012 annual meeting of the American Educational Research Association.
- Calvani, A. (2011). *Principi dell'istruzione e strategie per insegnare. Criteri per una didattica efficace*. Roma: Carocci.
- Calvani, A. (2014). *Come fare una lezione efficace*. Roma: Carocci.
- Calvani, A., Fini, A., & Ranieri, M. (2010). *La competenza digitale nella scuola*. Trento: Erickson.
- Capponi, M. (2009). *Un giocattolo per la mente. L'«informatica cognitiva» di Seymour Papert*. Perugia: Morlacchi.
- Carletti, A., & Varani, A. (2005). *Didattica costruttivista. Dalle teorie alla pratica in classe*. Trento: Erickson.

- Clements, D. H., & Battista, M. T. (1989). Learning of geometric concepts in a Logo environment. *Journal for Research in Mathematics Education*, 20, 450-467.
- Clements, D. H., & Battista, M. (1990, Novembre). The Effects of Logo on Children's Conceptualizations of Angle and Polygons. *Journal for Research in Mathematics Education*, 21(5), 356-371.
- Dewey, J. (1949). *Democrazia ed educazione*. Firenze: La nuova Italia
- Dewey, J. (1954). *Il mio credo pedagogico*. Firenze: La nuova Italia
- Dewey, J. (1973). *Scuola e società*. Firenze: La nuova Italia
- Downey, A., Elkner, J., & Meyers, C. (2002). *How to Think like a Computer Scientist: Learning with Python*. Wellesley (MA): Green Tea Press.
- Ferri, P. (2006). Le teorie e le epistemologie della formazione digitale. In P. Ferri, & S. Mantovani (A cura di), *Bambini e computer. Alla scoperta delle nuove tecnologie a scuola e in famiglia*. Milano: Etas.
- Ferri, P. (2011). *Nativi digitali*. Milano: Mondadori
- Formiconi, A. R. (2018). *Piccolo Manuale di LibreLogo. La Geometria della Tartaruga*.
- Fini, A. (2017). Nativi digitali. Un aggiornamento sulla discussione in rete. *Rivista Bricks*, 17 (1), 136-142.
- Ghezzi C. e Jazayeri M.(1994). *Concetti dei linguaggi di programmazione*. Milano: Franco Angeli
- Harel, I. & Papert, S. (1991) . *Constructionism*. Norwood, N.J.: Ablex

- Lewis, C. (2010). *How programming environment shapes perception, learning and goals: Logo vs. Scratch*. Proceeding SIGCSE '10 Proceedings of the 41st ACM technical symposium on Computer science.
- Lodi, M. (2014). *Imparare il pensiero computazionale, imparare a programmare*.
- Marchignoli, R., & Lodi, M. (2016). *EAS e pensiero computazionale. Fare coding nella scuola primaria*. Brescia: La Scuola.
- Ministero dell'Istruzione, dell'Università e della Ricerca. (2018). *Indicazioni Nazionali e Nuovi Scenari*.
- Ministero dell'Istruzione, dell'Università e della Ricerca. (2015). *Piano Nazionale Scuola Digitale*.
- Ministero dell'Istruzione, dell'Università e della Ricerca. (2018). *Indicazioni Nazionali e Nuovi Scenari*.
- Minsky, M., & Longo, G. (1990). *La società della mente*. Milano: Adelphi.
- Monacis, L., de Palo, V., & Sinatra, M. (2016) *Analisi dei fattori cognitivi-motivazionali determinanti l'uso integrato delle tecnologie nei docenti della scuola primaria e secondaria*, In Pace, R., Mangione, G., & Limone, P., *Dimensione didattica ,tecnologica e organizzativa* (p. 180-191) . Milano: FrancoAngeli
- Montessori, M. (1966). *La mente del bambino. Mente assorbente*. Milano: Garzanti.
- Moscetti, C. & Giordano, M. (2016). *Coding e pensiero computazionale nella Scuola Primaria*. Milano: La spiga.
- Nigris, E. (2007b). *Esperienza ed educazione*. In Nigris, E., Negri, S. C., & Zuccoli, F. *Esperienza e didattica. Le metodologie attive* (pp. 25-79). Roma: Carocci.

- Noss, R. (1987, Novembre). Children's Learning of Geometrical Concepts through Logo. *Journal for Research in Mathematics Education*, 18(5), 343-362.
- Papert, S. (1984). *Mindstorms: Bambini, computers e creatività*. Torino: Emme Edizioni.
- Papert, S. (1994). *I bambini e il computer*. Milano: Rizzoli.
- Papert, S. (1997). *Bambini e adulti a scuola con il computer*.
- Piaget, J. (1978). *Psicologia dell'intelligenza*. Firenze: Giunti-Barbera.
- Prenkys, M. (2001a). Digital Natives, Digital Immigrants. *On the Horizon*, 9(5), 1-6.
- Prensky, M. (2001b). Digital natives, digital immigrants, Part II: Do they really think differently? *On the Horizon*, 9(6), 15-24.
- Prensky, M. (2009). H. Sapiens Digital: From Digital Immigrants and Digital Natives to Digital Wisdom. *Innovate: Journal of Online*, 5(3)
- Ranieri, M. (2011). *Le insidie dell'ovvio. Tecnologie educative e critica della retorica tecnocentrica*. Pisa: ETS.
- Resnick, M. (2014). Give P's a Chance: Project, Peers, Passion, Play. *Constructionism and Creativity Conference*. opening keynote. Vienna
- Resnick, M (2005). *Some reflections on designing construction kits for kids*. Cambridge, Massachusetts: The MIT press.
- Simmons, M., & Cope, P. (1990, Agosto). Fragile Knowledge of Angle in Turtle Geometry. *Educational Studies in Mathematics*, 21(4), 375-382.
- Tapscott, D. (1998). *Growing Up Digital: The Rise of the Net Generation*. New York: McGraw-Hill.

- Weintrop, D., & Wilensky, U. (2015a). *Using commutative assessments to compare conceptual understanding in blocks and text base programs*. Proceeding ICER '15 Proceedings of the eleventh annual International Conference on International Computing Education Research, 101-110.
- Weintrop, D., & Wilensky, U. (2015b). *To block or not to block, that is the question: Students' perceptions of block-based programming*. Proceeding IDC '15 Proceedings of the 14th International Conference on Interaction Design and Children, 199-208.
- Wing, J. M. (2006). Computational Thinking. *Communications of the ACM*, 49(3), 33-35.

Sitografia

- <http://codemooc.org/mooc/> (consultato il 13/08/2018)
- <http://codeweek.it/cody-roby/> (consultato il 21/08/2018)
- <http://codeweek.it/lm05-codyway/> (consultato il 21/08/2018)
- <http://www.coderdojoitalia.org/> (consultato il 13/08/2018)
- http://www.miur.gov.it/documents/20182/0/DM+741_2017.pdf/f7768e43-fb00-447d-8f27-8f4f584f2f8f?version=1.0 (consultato il 10/08/2018)
- <http://www.programmailfuturo.it> (consultato il 11/08/2018)
- <http://www.treccani.it/vocabolario/linguaggio/> (consultato il 02/09/18)
- <https://code.org/> (consultato il 12/08/2018)
- <https://programmailfuturo.it/media/docs/Descrizione-progetto-Programma-il-Futuro.pdf> (consultato il 11/08/2018).
- <https://programmailfuturo.it/progetto/monitoraggio-del-progetto> (consultato il 11/08/2018)
- <https://programmailfuturo.it/progetto/perche-partecipare/informatica-e-scuola> (consultato in data 11/08/2017).
- <https://scratch.mit.edu/>(consultato il 26/08/18)
- <https://www.campustore.it/robotica-educativa-elettronica-coding/bee-bot-blue-bot/bee-bot-ricaricabile.html> (consultato il 20/08/2018)
- <https://www.campustore.it/robotica-educativa-elettronica-coding/bee-bot-blue-bot/blue-bot-13.html> (consultato il 20/08/2018)
- <https://www.campustore.it/robotica-educativa-elettronica->

[coding/cubetto.html](#) (consultato il 20/08/2018)

- <https://www.campustore.it/robotica-educativa-elettronica-coding/dash-dot.html> (consultato il 20/08/2018)
- <https://www.campustore.it/robotica-educativa-elettronica-coding/lego-education/mindstorms-ev3.html> (consultato il 20/08/2018)
- <https://www.campustore.it/robotica-educativa-elettronica-coding/mbot-makeblock/mbot.html> (consultato il 20/08/2018)
- <https://www.gov.uk/government/publications/national-curriculum-in-england-computing-programmes-of-study/national-curriculum-in-england-computing-programmes-of-study> (consultato il 10/08/2018)

Appendice

a) Test 1

Nome e Cognome.....Data.....

TEST DI GEOMETRIA N°1

1) Cos'è un angolo? Disegnane uno.

.....
.....

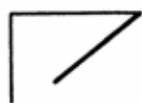
2) Puoi disegnare un angolo maggiore rispetto a quello che hai disegnato prima?
Perché è maggiore?

.....
.....

3) Cerchia tra queste figure quelle che rappresentano angoli.



4) Quanti angoli hanno queste figure? Scrivi il numero di angoli sotto ad ogni figura.

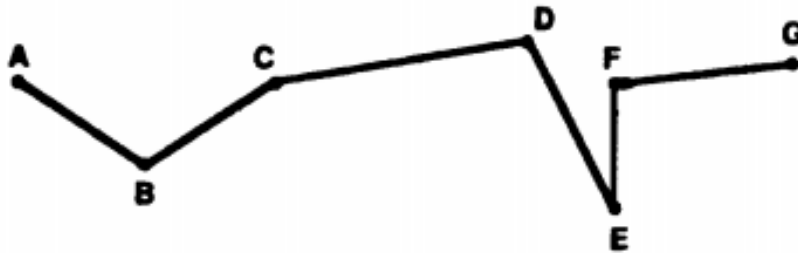


..... Angoli



..... Angoli

5) Stai camminando su questo sentiero. Inizi dal punto A e finisci nel punto G. In quale punto devi ruotare di più? Indicalo con una freccia di colore blu. In quale punto devi ruotare di meno? Indicalo con una freccia di colore rosso.



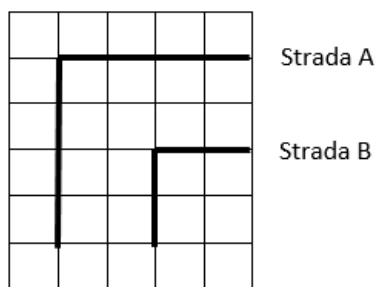
6) Un robot ruota di 90 gradi ogni volta che ruota. Quante rotazioni deve fare il robot per puntare nella stessa direzione iniziale? Perché?

.....

7) Un robot ruota di 30 gradi ogni volta. Quante rotazioni deve fare prima di puntare la stessa direzione iniziale? Perché?

.....

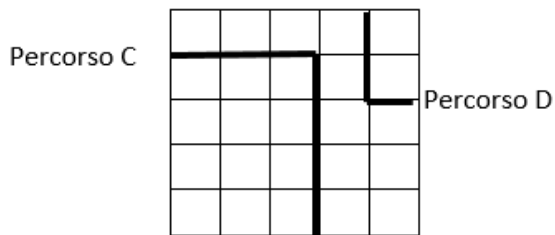
8) Nella città chiamata Quadrata tutte le strade sono disposte su uno schema quadrato. Ci sono solo due strade principali, che sulla mappa sono rappresentate da linee più scure.



Seleziona la risposta che pensi sia corretta:

- La strada A curva di più della strada B
- La strada B curva di più della strada A
- La strada A e la strada B curvano nello stesso modo
- Non puoi dirlo

9) Nella città di Quadrata ci sono due nuove strade C e D



Seleziona la risposta che pensi sia corretta:

- La strada C curva più della strada D
- La strada D curva più della strada C
- Le strade C e D curvano nello stesso modo
- Non puoi dirlo

10) Osserva gli angoli A e B di questi due triangoli.



Seleziona la risposta che pensi sia corretta:

- L'angolo A è maggiore dell'angolo B
- L'angolo B è maggiore dell'angolo A
- Gli angoli A e B hanno la stessa ampiezza
- Non puoi dirlo

11) In ognuno di questi triangoli tutti i lati hanno la stessa lunghezza.



Seleziona la risposta che pensi sia corretta:

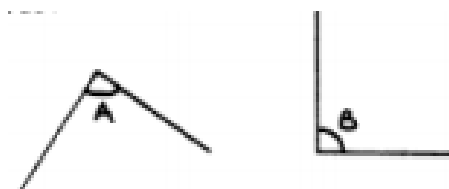
L'angolo C è maggiore dell'angolo D

L'angolo D è maggiore dell'angolo C

Gli angoli C e D hanno la stessa ampiezza

Non puoi dirlo

12) Osserva gli angoli A e B



Seleziona la risposta che pensi sia corretta:

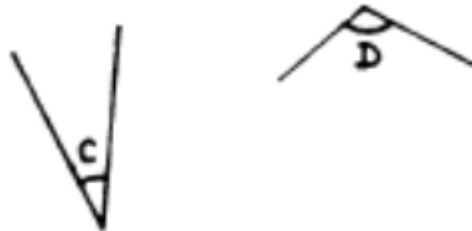
L'angolo A è maggiore dell'angolo B

L'angolo B è maggiore dell'angolo A

Gli angoli A e B hanno la stessa ampiezza

Non puoi dirlo

13) Osserva gli angoli



Seleziona la risposta che pensi sia corretta:

L'angolo C è maggiore dell'angolo D

L'angolo D è maggiore dell'angolo C

Gli angoli C e D hanno la stessa ampiezza

Non puoi dirlo

b)Test 2

Nome e Cognome.....Data.....

TEST DI GEOMETRIA N°2

1) Cos'è un angolo? Disegnane uno.

.....
.....

2) Puoi disegnare un angolo maggiore rispetto a quello che hai disegnato prima?
Perché è maggiore?

.....
.....

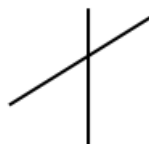
3) Cerchia tra queste figure quelle che rappresentano angoli.



4) Quanti angoli hanno queste figure? Scrivi il numero di angoli sotto ad ogni figura.



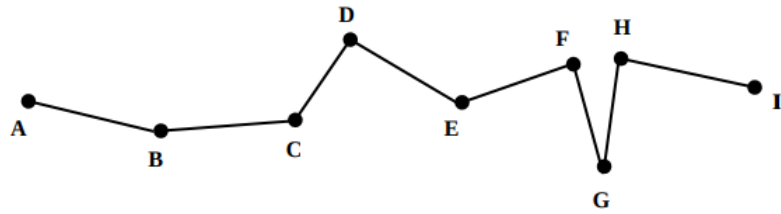
..... Angoli



..... Angoli

5) Stai camminando su questo sentiero. Inizi dal punto A e finisci nel punto I. In quale punto devi ruotare di più? Indicalo con una freccia di colore blu. In quale

punto devi ruotare di meno? Indicalo con una freccia di colore rosso.



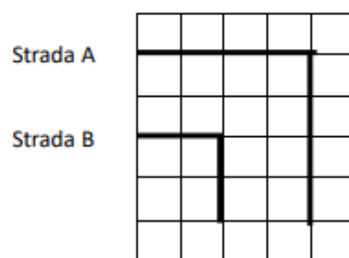
6) Un robot ruota di 60 gradi ogni volta che ruota. Quante rotazioni deve fare il robot per puntare nella stessa direzione iniziale? Perché?

.....

7) Un robot ruota di 90 gradi ogni volta che ruota. Quante rotazioni deve fare il robot per puntare nella direzione opposta? Perché?

.....

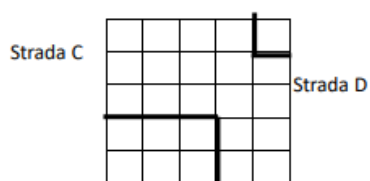
8) Nella città chiamata Quadrata tutte le strade sono disposte su uno schema quadrato. Ci sono solo due strade principali, che sulla mappa sono rappresentate da linee più scure.



Seleziona la risposta che pensi sia corretta:

- La strada A curva di più della strada B
- La strada B curva di più della strada A
- La strada A e la strada B curvano nello stesso modo
- Non puoi dirlo

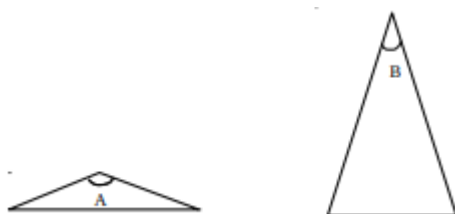
9) Nella città di Quadrata ci sono due nuove strade C e D



Seleziona la risposta che pensi sia corretta:

- La strada C curva più della strada D
- La strada D curva più della strada C
- Le strade C e D curvano nello stesso modo
- Non puoi dirlo

10) Osserva gli angoli A e B di questi due triangoli.



Seleziona la risposta che pensi sia corretta:

- L'angolo A è maggiore dell'angolo B
- L'angolo B è maggiore dell'angolo A

Gli angoli A e B hanno la stessa ampiezza

Non puoi dirlo

11) In ognuno di questi triangoli tutti i lati hanno la stessa lunghezza.



Seleziona la risposta che pensi sia corretta:

L'angolo C è maggiore dell'angolo D

L'angolo D è maggiore dell'angolo C

Gli angoli C e D hanno la stessa ampiezza

Non puoi dirlo

12) Osserva gli angoli A e B



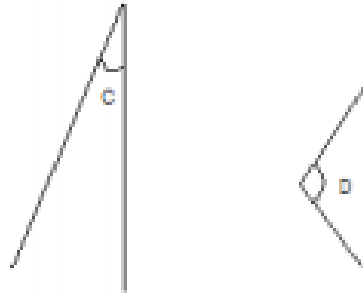
Seleziona la risposta che pensi sia corretta:

L'angolo A è maggiore dell'angolo B

L'angolo B è maggiore dell'angolo A

Gli angoli A e B hanno la stessa ampiezza

Non puoi dirlo



13) Osserva gli angoli

Seleziona la risposta che pensi sia corretta:

L'angolo C è maggiore dell'angolo D

L'angolo D è maggiore dell'angolo C

Gli angoli C e D hanno la stessa ampiezza

Non puoi dirlo

c)Test 3

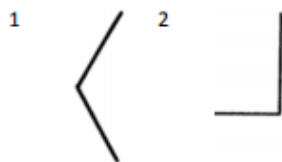
Nome e Cognome.....Data.....

TEST DI GEOMETRIA N°3

1) Cos'è un angolo? Disegnane uno.

.....
.....

2) Quale tra i due angoli è il maggiore? Perché?



.....
.....

3) In ogni coppia di figura, individua l'angolo e cerchiolo



4) Quanti angoli hanno queste figure? Scrivi il numero di angoli sotto ad ogni figura.



5)

Questo è il sentiero che il robot ha seguito per arrivare dal punto A al punto G. Adesso prova tu a disegnare un sentiero che inizi nel punto A e finisca nel punto G. Ad ogni cambio di direzione scrivi una lettera. Indica con una freccia di colore blu il punto in cui devi ruotare di più e con una freccia di colore rosso il punto in cui devi ruotare di meno.

A.

.G

6) Un robot ruota di 120 gradi ogni volta che ruota. Quante rotazioni deve fare il

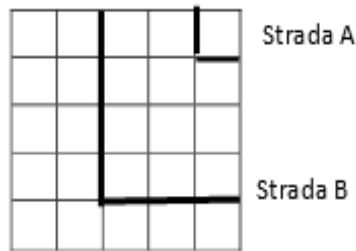
robot per puntare nella stessa direzione iniziale? Perché?

.....
.....
.....

7) Un robot ruota di 45 gradi ogni volta che ruota. Quante rotazioni deve fare il robot per puntare nella direzione opposta? Perché?

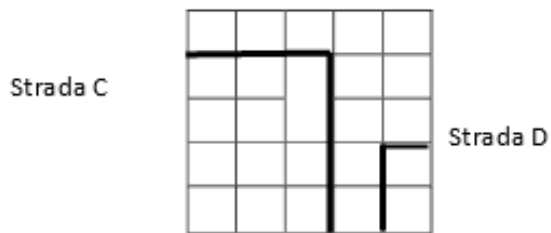
.....
.....
.....

8) Nella città chiamata Quadrata tutte le strade sono disposte su uno schema quadrato. Ci sono solo due strade principali, che sulla mappa sono rappresentate da linee più scure.



Seleziona la risposta che pensi sia corretta:

- La strada A curva di più della strada B
- La strada B curva di più della strada A
- La strada A e la strada B curvano nello stesso modo
- Non puoi dirlo

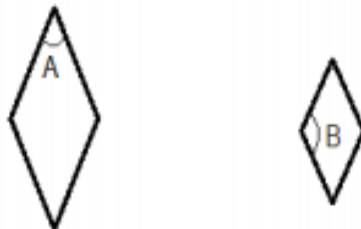


9) Nella città di Quadrata ci sono due nuove strade C e D

Seleziona la risposta che pensi sia corretta:

- La strada C curva più della strada D
- La strada D curva più della strada C
- Le strade C e D curvano nello stesso modo
- Non puoi dirlo

10) Osserva gli angoli A e B di questi rombi



Seleziona la risposta che pensi sia corretta:

- L'angolo A è maggiore dell'angolo B
- L'angolo B è maggiore dell'angolo A
- Gli angoli A e B hanno la stessa ampiezza
- Non puoi dirlo

11) In ognuno di questi triangoli tutti i lati hanno la stessa lunghezza.



Seleziona la risposta che pensi sia corretta:

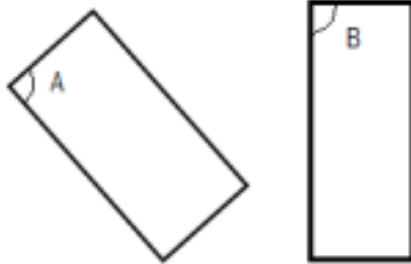
L'angolo C è maggiore dell'angolo D

L'angolo D è maggiore dell'angolo C

Gli angoli C e D hanno la stessa ampiezza

Non puoi dirlo

12) Osserva gli angoli A e B



Seleziona la risposta che pensi sia corretta:

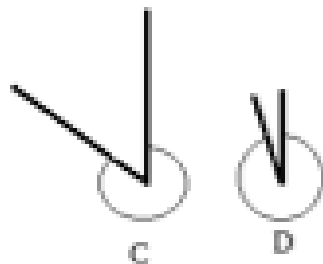
L'angolo A è maggiore dell'angolo B

L'angolo B è maggiore dell'angolo A

Gli angoli A e B hanno la stessa ampiezza

Non puoi dirlo

13) Osserva gli angoli



Seleziona la risposta che pensi sia corretta:

L'angolo C è maggiore dell'angolo D

L'angolo D è maggiore dell'angolo C

Gli angoli C e D hanno la stessa ampiezza

Non puoi dirlo

d) Librino con i comandi scritti dai bambini e alcune delle loro creazioni con Logo e Scratch.

Programmiamo con Logo e SCRATCH



Classe 5° A

RIGHT
COMANDI DI LOGO
LEFT

HOME

REPEAT



BACK

HOME

LA TARTARUGA TORNA NEL PUNTO DI PARTENZA CON LA TESTA RIVOLTA VERSO L'ALTO.

CLEARSCREEN

CANCELLA LE RIGHE DAL FOGLIO

FORWARD

LA TARTARUGA SI MUOVE IN AVANTI DI

BACK

LA TARTARUGA TORNA INDIETRO DI

LEFT

LA TARTARUGA RUOTA A SINISTRA DI

RIGHT ...

LA TARTARUGA RUOTA A DESTRA DI

**PENCOLOR
"PINK"**

CAMBIA IL COLORE DELLA PENNA
ES. PENCOLOR "GREEN"

**FILL COLOR
"PURPLE" FILL**

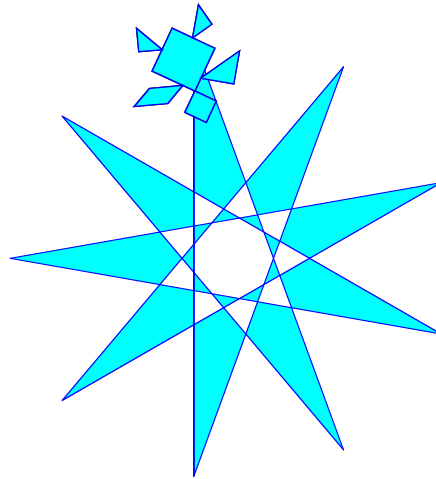
COLORA LA FIGURA
ES. FILL COLOR "RED" FILL

REPEAT [4]

PER RIPETERE I COMANDI SCRITTI NELLE PARENTESI.

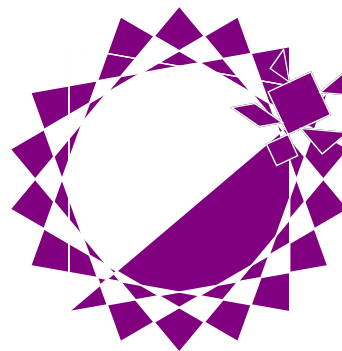
LA NOSTRA STELLA

```
HOME  
CLEARSCREEN  
PENCOLOR "BLUE"  
REPEAT 10[  
FORWARD 200  
RIGHT 160  
]  
FILLCOLOR "AQUA" FILL
```



SUPER FIGURA

```
HOME  
CLEARSCREEN  
PENCOLOR "WHITE"  
REPEAT 20 [  
FORWARD 100  
RIGHT 100  
]  
FILLCOLOR "PURPLE" FILL
```



COMANDI DI SCRATCH

quando si clicca su 

QUANDO SI CLICCA SULLA BANDIERINA VERDE I COMANDI SOTTO QUESTO BLOCCO VENGONO ESEGUITI

penna giù

IL GATTO LASCIA UN SEGNO QUANDO SI MUOVE

fai 10 passi

IL GATTO SI MUOVE IN AVANTI DI 10 PASSI

ruota di  15 gradi

IL GATTO RUOTA A DESTRA DI 15° GRADI

ruota di  15 gradi

IL GATTO RUOTA A SINISTRA DI 15° GRADI

penna su

IL GATTO NON LASCIA UN SEGNO QUANDO SI MUOVE

pulisci

CANCELLA I SEGNI DELLA PENNA

vai a x: 0 y: 0

SONO LE COORDINATE DELLA POSIZIONE DEL GATTO

punta in direzione 90 

DEFINISCE LO SPESORE DELLA PENNA

usa penna di colore 

DEFINISCE IL COLORE DELLA PENNA

nascondi

IL GATTO SPARISCE

IMPOSTAZIONI BASE PER LAVORARE



LA MIA INIZIALE





UNIVERSITÀ
DEGLI STUDI
FIRENZE

Scuola di
Studi Umanistici
e della Formazione

Corso di Laurea in
Scienze della Formazione Primaria N.O.

Relazione finale di tirocinio

Tirocinante

Fabiola Izzo

Tutor scolastici

Rossana Cercenà

Arianna Garuzzo

Tutor universitari

Lucia Maddii

Indice

Introduzione	3
Capitolo 1 – L’esperienza del tirocinio diretto	4
1.1 Le sedi del tirocinio.....	4
1.2 Il tirocinio nella scuola dell’infanzia.....	4
1.3 Il tirocinio nella scuola primaria	13
Capitolo 2- Valutazioni di fine percorso	24
2.1 La scuola come comunità professionale	24
2.2 Formazione professionale conseguita	25
Conclusioni	30
Bibliografia	31
Sitografia	31
Appendice	32

Introduzione

Le attività di tirocinio sono state fondamentali durante il mio percorso di formazione come insegnante. Dalle prime osservazioni fino alla realizzazione di alcune mie lezioni, supportata da un percorso di tirocinio indiretto che mi ha fornito le basi per poter iniziare ad avere un ruolo attivo in classe, ripercorro in questa relazione questi quattro anni di tirocinio svolto inizialmente con molto timore ma con tanta passione e voglia di migliorare.

Fondamentali sono stati i rapporti con le persone che mi hanno seguita durante questo percorso, in particolare le tutor che mi hanno guidata e sostenuta, ma soprattutto che mi hanno aiutata ad acquisire una professionalità di base.

Il primo capitolo ripercorre le esperienze vissute nella scuola dell'infanzia e nella scuola primaria dell'istituto comprensivo che mi ha accolto, descrivendo i diversi ambienti, ma soprattutto incentrandosi sugli strumenti, le metodologie, le pratiche osservate e le attività svolte.

Il secondo capitolo inizia con un'analisi delle attività collegiali alle quali ho partecipato. Questi momenti mi hanno permesso di instaurare relazioni sia con il personale scolastico che con i genitori. Un paragrafo viene poi dedicato all'analisi del percorso di tirocinio indiretto. Il capitolo si conclude con un bilancio complessivo dell'esperienza e una riflessione sulle competenze professionali sviluppate.

Nelle conclusioni termino con una breve riflessione su come questa esperienza ha cambiato il mio modo di essere oggi.

Capitolo 1

L'esperienza del tirocinio diretto

1.1 Le sedi del tirocinio

Ho svolto il mio percorso di tirocinio nell'Istituto Comprensivo Lucca 2 situato nella periferia di Lucca a pochi passi dal centro storico. I plessi presso cui ho svolto il tirocinio sono la scuola dell'infanzia "San Concordio Centrale" e la scuola primaria "C.L. Collodi" situate in due strutture adiacenti. Ho inizialmente scelto questo istituto poiché è situato vicino alla mia abitazione, ma ho deciso di continuare il tirocinio in questo istituto poiché, durante il primo anno di tirocinio, ho avuto modo di scoprire le diverse attività che l'istituto offriva che ho ritenuto interessanti e stimolanti, ma principalmente per la possibilità di continuare a lavorare con la mia tutor scolastica che, con il tempo, è diventata per me un modello di riferimento.

Da un lato questa scelta mi ha penalizzato perché, nonostante abbia visto classi diverse, non ho potuto osservare istituti differenti tra di loro; dall'altro però, soprattutto nella scuola primaria, ho avuto la possibilità di seguire una classe quasi per l'intero percorso diventandone un membro e conoscendo bene i bambini. L'istituto è il più grande di Lucca e comprende diversi plessi tra cui: quattro scuole dell'infanzia, quattro scuole primarie e una scuola secondaria di primo grado. Il numero di iscritti a tale istituto è elevato e questo determina, come emerge dal documento RAV, la presenza di una popolazione scolastica variegata dal punto di vista socio-economico e culturale, registrando un livello medio-basso dell'ESCS.

1.2 Il tirocinio nella scuola dell'infanzia

La scuola dell'infanzia "San Concordio Centrale" è divisa in tre sezioni miste composte ognuna da circa ventotto bambini: le api, i bruchi e le coccinelle. Le tre sezioni condividono un grande salone con diversi giochi strutturati, un angolo biblioteca con uno spazio morbido dove poter leggere, alcuni tavolini dove svolgere diversi lavori e uno spazio con alcuni strumenti musicali. Questo spazio comune viene utilizzato per diverse attività come l'accoglienza, la musicoterapia e

l'attività motoria. La scuola dispone anche di un ampio refettorio, dove tutte le sezioni pranzano insieme e un grande giardino, accessibile da tutte le sezioni, che circonda l'edificio e dispone di diversi giochi.

Ogni sezione è poi divisa in tre stanze: la sala principale, la stanza dell'accoglienza e i bagni.

La sala principale è organizzata in angoli: angolo delle costruzioni e del movimento, angolo della cucina e dei travestimenti, angolo della pittura e angolo della lettura. Al centro della stanza troviamo diversi tavolini divisi in tre gruppi in modo da poter anche lavorare dividendo i bambini per età.

Una porta collega la stanza principale con quella dell'accoglienza, questa stanza dispone di diverse panchine che possono essere messe in cerchio, e quindi viene utilizzata per le attività di routine e per tutti i momenti di preparazione: prima della mensa, prima di uscire o prima di svolgere un'attività. È importante per i bambini avere degli spazi specifici dove svolgere le attività di routine, in questo modo associano quella stanza a determinate attività.

Come per la stanza principale, le pareti sono piene di cartelloni funzionali per lo svolgimento delle attività; vi è il tabellone delle presenze, delle barchette per ogni giorno della settimana, alle quali vengono attaccati i contrassegni dei bambini che sono titolari per quel giorno dei diversi incarichi, il tabellone dei contrassegni, il tabellone per segnare il tempo, la filastrocca dei giorni della settimana, la linea delle attività della giornata, la linea dei numeri, un cartellone con le regole condivise dai bambini della sezione e infine, su un lato della stanza, vi è un grosso cartellone con disegnato un albero di cartone che viene decorato con i disegni dei bambini a seconda della stagione.

In fondo a questa stanza troviamo un altro piccolo angolo biblioteca con un tavolino, questo ambiente viene poco utilizzato dai bambini, anche se nello spazio comune la biblioteca viene usata maggiormente, mentre potrebbe essere utile uno spazio morbido, che nell'aula manca, e potrebbe sostituire il tavolino che non viene utilizzato. Da questa stanza si accede al bagno che è ben organizzato ed è a misura di bambino. L'organizzazione degli spazi è fondamentale nella scuola dell'infanzia

L'organizzazione degli spazi e dei tempi diventa elemento di qualità pedagogica dell'ambiente educativo e pertanto deve essere oggetto di esplicita progettazione e verifica. (Ministero dell'Istruzione, dell'Università e della Ricerca, 2012).

Per aiutare i bambini ad organizzarsi durante il gioco libero e non creare eccessiva confusione in un solo angolo un'insegnante ha introdotto lo scorso anno dei cartellini che indicano quanti bambini possono stare in ogni angolo. Osservando le attività svolte nella scuola dell'infanzia, ho compreso l'importanza di un'attenta progettazione da parte dell'insegnante. Ho avuto la possibilità di osservare diverse insegnanti, ognuna con un diverso metodo e una diversa gestione delle attività, alcune lasciavano molto spazio al gioco libero e le attività erano spesso poco strutturate, altre, invece, organizzavano diverse attività nel corso della giornata, preparando in anticipo molti materiali, anche per i momenti di gioco libero, durante il quale le insegnanti fornivano diverse proposte. Le due diverse tipologie di insegnamento influenzavano fortemente il clima della classe e il diverso sviluppo dei bambini: chi è abituato a molto gioco libero risulta più autonomo ma spesso è difficile che svolga con continuità le attività strutturate e anche le routine sono poco seguite, mentre, chi svolge più attività strutturate è abituato a seguire il lavoro e a rispettare determinate routine, quindi, è possibile strutturare anche attività più complesse e lunghe. Un altro fattore che influenza il tipo di clima e lo svolgimento delle attività, è il tipo di rapporto che regola il lavoro del team docente, è necessaria collaborazione nella progettazione dei percorsi didattici ed è importante che le insegnanti condividano le regole per non creare confusione nel bambino. Fondamentali nella scuola dell'infanzia sono le routine che permettono ai bambini di orientarsi nella giornata, fornendo loro anche sicurezza su quello che succederà e permettono di sviluppare maggiore autonomia durante le attività realizzate nel corso della mattinata. Una delle attività di routine più importanti è quella che si svolge la mattina. Durante questi anni ho inizialmente solo osservato queste attività ma negli ultimi due anni spesso ho avuto la possibilità di gestirle. Nell'ultimo anno di tirocinio una nuova maestra ha accompagnato la mia tutor introducendo nuove attività di routine molto

interessanti. Per svolgere queste attività per prima cosa i bambini si spostano nella stanzina più piccola, dove vengono disposte le panchine a cerchio; prima di iniziare le attività vengono dedicati alcuni momenti al dialogo in modo che i bambini possano raccontare qualcosa che vogliono condividere. Poi iniziano le attività, solitamente la prima consiste nel far ripetere ad alcuni bambini la filastrocca sui giorni della settimana davanti alle immagini che riprendono tale filastrocca, arrivando al giorno nel quale si trovano (allegato 1) .

Poi l'insegnante sceglie tre bambini che saranno i "maestri", questi bambini hanno il compito di prendere i contrassegni e chiamare uno per volta gli altri compagni a descrivere e poi attaccare il loro segmino al giorno giusto. L'insegnante ha creato un grande cartellone dalla forma circolare diviso in sette sezioni di colore differente, ognuna per un giorno della settimana, al centro è posizionata una freccia che viene spostata ogni giorno. I bambini sono tre poiché ognuno è incaricato di chiamare i bambini di un gruppo di età. I bambini danno molta importanza al contrassegno che non solo li rappresenta nella routine, ma li aiuta anche ad avere chiari i loro spazi all'interno dell'aula: hanno il loro simbolo sopra l'appendipanni esterno, nel bagno, nello spazio del mobile centrale, dove inseriscono i loro disegni e i loro effetti personali e anche il cartellino con il loro nome è contrassegnato dal simbolo. Il contrassegno cambia ogni giorno e dimostra anche il fatto che sono cresciuti, inoltre per il bambino è importante avere chiari gli spazi e capire quali sono i suoi e gli altrui. L'uso del contrassegno è stato introdotto nella scuola dell'infanzia dalla fine dell'ottocento dalle sorelle Rosa e Carolina Agazzi ed ha come obiettivi quelli di favorire il mantenimento dell'ordine della classe e lo sviluppo della cura verso i propri oggetti e quelli degli altri facendo attenzione durante il loro uso e nel riporli. In questo modo si sviluppa, anche dal punto di vista dell'educazione sociale, l'assunzione di responsabilità verso le cose che appartengono a se stesso e agli altri.

Sono quindi simboli che Rosa Agazzi (1966) definisce "*coefficienti dell'ordine materiale*" utili per "*agevolare le quotidiane operazioni di uso di oggetti, in riferimento agli indumenti come il grembiule, il bavagliolo, l'asciugamano, agli arredi come l'armadietto personale e agli oggetti come la "scatola individuale" in cui il bambino può riporre i disegni, gli oggetti portati da casa e le*

“cianfrusaglie”.

Dopo il momento delle presenze vengono assegnati gli incarichi del giorno (chiudi fila, contare i maschi, osservare il tempo, eccetera) e attaccati su un cartellone apposito contenente il contrassegno e il nome del bimbo.

Queste attività sono importanti per sviluppare maggiore autonomia e responsabilità, perché il bambino assume un impegno che deve portare a termine. Svolgendo questi compiti imparano anche ad aspettare il proprio turno e affinano competenze in diversi ambiti. Le successive attività di calendario che vengono svolte, ad esempio, sviluppano le competenze temporali poiché permettono ai bambini di familiarizzare in maniera progressiva con il passare del tempo, di memorizzare la successione dei giorni della settimana, di imparare a distinguere tra ieri oggi e domani, scoprono il susseguirsi dei mesi e delle diverse stagioni e affinano le capacità di osservare gli eventi atmosferici. Inoltre queste attività sviluppano competenze logiche e matematiche poiché imparano a confrontare, contare, associare e simbolizzare. Oltre alla filastrocca vengono svolte altre attività di calendario, la principale viene svolta con una linea dei giorni del mese e tre mollette con delle frecce applicate; viene chiesto ai bambini di indicare il giorno e poi la maestra posiziona la freccia sul numero indicato. Dopo viene chiesto ai bambini “domani che giorno sarà?” e “ieri che giorno era?” e anche in questi casi vengono posizionate le relative frecce. (allegato 1)

Un'altra attività che aiuta lo sviluppo di competenze matematiche è quella di contare i bambini presenti: prima vengono contate le bambine, poi i bambini ed infine tutti insieme. Anche questa attività è supportata da una linea con i numeri sulla quale vengono posizionate delle frecce sul numero corretto, viene anche chiesto ad alcuni bambini di trovare il numero uguale a quello segnato sulla linea tra diversi cartellini e di attaccarlo sotto il simbolo corretto. (Allegato 1)

Le attività svolte durante la mattina nelle ore di compresenza sono quelle che vengono maggiormente strutturate, essendo sezioni miste è necessario preparare bene i materiali in modo da assegnare ad ogni gruppo un lavoro adatto a loro e, solitamente per i bambini di cinque anni deve essere in parte autonomo. Le maestre, infatti, spesso si dividevano e una gestiva il lavoro dei bambini di tre anni, mentre l'altra si dedicava ai bambini di quattro e di cinque anni. Avendo

avuto la possibilità di vedere una sezione mista ho osservato gli aspetti positivi e negativi di questa tipologia di sezione: sicuramente avere la possibilità di confrontarsi con età differenti è stimolante per i bambini. Spesso ho visto i più piccoli cercare di imitare i grandi e quindi provare nuove attività che altrimenti non avrebbero svolto; inoltre, nel momento della separazione dai genitori, i bambini più grandi sono spesso fondamentali per tranquillizzare i più piccoli, questo, aiuta anche i più grandi che si responsabilizzano. Questo rapporto tra grandi e piccoli è stato anche incoraggiato dalle maestre che hanno assegnato ad ogni bimbo di tre anni un bambino di cinque per fargli da tutor, questo bambino doveva aiutare il più piccolo durante lo svolgimento delle attività di routine come ad esempio lavarsi le mani o versare l'acqua nel bicchiere a mensa. Però, come espresso nelle righe precedenti, una sezione mista richiede una programmazione e una preparazione dei materiali ancora più attenta, sia per diversificare le attività sia per programmare strategie per lo svolgimento di quelle in comune, in modo da coinvolgere tutti i bambini. Alcune esperienze, proprio per stimolare al meglio i bambini con attività adeguate al loro sviluppo e per una gestione migliore del gruppo, venivano svolte per età: i bambini si spostavano nelle tre sezioni e ognuna gestiva una fascia di età, anche le insegnanti si spostavano tra le sezioni. Tra queste molto interessanti sono stati i laboratori scientifici che ogni anno propongono esperienze che avvicinano i bambini all'ambiente nel quale vivono. Quest'anno il tema principale era la terra, io ho potuto seguire e aiutare nella gestione del percorso con i bambini di cinque anni, principalmente ho aiutato durante il lavoro conclusivo dopo l'attività di osservazione dei lombrichi che hanno trovato nella terra. I bambini dovevano disegnare un lombrico, avevano anche la possibilità di osservarlo nuovamente, al centro del foglio che era stato preparato in precedenza con uno spazio apposito; poi descrivevano il lombrico per come lo avevano percepito (lungo, marrone, liscio, ad anelli). Quando i bambini trovavano un aggettivo io disegnavo una linea con scritto quello che avevano detto e alla fine del segmento il bambino incollava uno dei materiali, precedentemente preparati, che associava alla caratteristica descritta. Durante questa attività i bambini avevano la possibilità di descrivere l'esperienza vissuta ma anche di fissarla grazie all'utilizzo del disegno e dei materiali che gli

permettono di ricordare le sensazioni descritte. Ho sempre osservato una particolare attenzione verso la rappresentazione delle attività svolte, tramite rappresentazioni grafiche, foto o altre produzioni con materiali come paste modellabili, momento importante valorizzato anche dalle “Indicazioni nazionali per il curricolo della scuola dell’infanzia e del primo ciclo di istruzione”:

La pratica della documentazione cartacea e fotografica va intesa come processo che produce tracce, memoria e riflessione negli adulti e nei bambini, rendendo visibili le modalità i percorsi di formazione e permettendo di apprezzare i progressi dell’apprendimento individuale e di gruppo. (Ministero dell’Istruzione, dell’Università e della Ricerca, 2012)

Durante questi anni ho anche potuto osservare buone pratiche sulla gestione dei momenti di confusione, un aspetto che mi preoccupava molto: spesso vengono utilizzate delle canzoncine abbinate a specifici movimenti che diminuiscono sempre più il tono fino a diventare silenziose; anche creare dei giochi del silenzio spesso è efficace o cercare di avere il contatto visivo con ogni singolo bambino.

Ritengo inoltre molto interessante e utile strutturare le attività di accoglienza a partire da un libro, oltre ad avvicinare i bambini ad una storia, oggi esistono molti libri con valore educativo, storie che trattano di argomenti importanti per lo sviluppo dei bambini come l’amicizia, la scoperta di sé o valorizzare le differenze. In questo tipo di libro il protagonista accompagnerà i bambini per tutto il percorso diventando un punto fisso per i più piccoli e il coinvolgimento con la storia li stimolerà a svolgere le attività. Durante l’ultimo anno scolastico al quale ho partecipato la storia scelta è stata “Pezzettino” di Leo Lionni che aveva come obiettivo quello di accompagnare i bambini alla scoperta della propria identità, aiutandoli a comprendere il senso di appartenenza ad un gruppo, ad una comunità in cui sentirsi riconosciuti nella propria unicità. Negli anni precedenti sono state utilizzate “Piccolo blu e piccolo giallo” di Leo Lionni e “Elmer, l’elefante variopinto” di David McKee. L’uso dei libri è utile anche durante le attività di fine giornata quando i bambini rimasti delle tre

sezioni si riuniscono in una sola stanza. La maestra spesso ha dedicato questo spazio alla lettura di una storia breve seguita da una piccola attività, ad esempio, dopo la lettura di una storia sui mostri ha chiesto ad ogni bambino di disegnare con dei gessetti colorati sul pavimento dei mostri, e poi con il libro “No, no e poi no” di Mireille d’Allancé hanno utilizzato sempre i gessetti per scrivere “No” e insieme riflettere sulle situazioni durante le quali vengono detti.

Di seguito descrivo la prima attività che ho condotto in totale autonomia:

Le case di piccolo blu e di piccolo giallo

Ho svolto questa attività durante il terzo anno di tirocinio. L’attività è rivolta ai bambini di tre anni ed è stata svolta durante la mattina dopo le attività di routine. Mi era stato chiesto di preparare un’attività legata al libro che la maestra stava leggendo per il percorso di accoglienza, “Piccolo blu e piccolo giallo” di Leo Lionni. Il lavoro sulla storia era solamente all’inizio e i bambini avevano svolto un’attività sui due personaggi principali, abbiamo scelto insieme di incentrare invece questa attività sulla casa dei protagonisti e sulle loro famiglie.

Allegato 2

Campi di esperienza

- Il sé e l’altro.
- Immagini, suoni e colori.

Traguardi per lo sviluppo delle competenze relativi al campo di esperienza “Il sé e l’altro”:

- Sa di avere una storia personale e familiare, conosce le tradizioni della famiglia, della comunità e le mette a confronto con altre.
- Riflette, si confronta, discute con gli adulti e con gli altri bambini e comincia a riconoscere la reciprocità di attenzione tra chi parla e chi ascolta.

Traguardi per lo sviluppo delle competenze relativi al campo di esperienza
“immagini, suoni e colori”:

- Esprime storie attraverso il disegno, la pittura e altre attività manipolative; utilizza materiali e strumenti, tecniche espressive e creative;

Obiettivi didattici

- Rappresentare graficamente la storia
- Identificare alcune caratteristiche dei personaggi: forma, dimensione e colore.

L'attività è stata svolta dopo la rilettura della storia di Leo Lionni; ho spiegato il lavoro, preparato dall'altra maestra ai gruppi dei bambini di quattro e cinque anni, esso consisteva nel colorare una collana con i due colori dei protagonisti alternandoli. Dopo, sono rimasta con il gruppo dei tre anni, ho ripreso il libro focalizzando l'attenzione sulle abitazioni e sui membri della famiglia presenti nella casa. Dall'osservazione delle immagini è nata una riflessione durante la quale i bambini hanno espresso la loro visione della casa e della loro famiglia. Poi ho spiegato loro quello che saremmo andati a fare e i bambini si sono spostati sul piano di lavoro. Poiché lavoravano con le tempere, per prima cosa è stato necessario prepararli aiutandoli a mettere il grembiule e a posizionarsi, questa è un'attività nella quale bisogna ben tenere conto dei tempi.

Ogni bambino aveva a disposizione il materiale per il lavoro; essendo i bambini di tre anni ho preparato in precedenza della carta da pacchi riprendendo la forma delle case disegnate sul libro e alcuni contenitori con della tempera blu e gialla. I barattoli per le tempere erano molti di meno rispetto ai bambini, questo anche per aiutarli ad imparare il rispetto del proprio turno. Ogni bambino ispirandosi ai disegni del libro doveva riprodurre piccolo blu e piccolo giallo con i loro genitori in due cassette diverse.

1.3 Il tirocinio nella scuola primaria

La classe dove ho svolto per il maggior numero di ore il tirocinio è molto piccola ed è posizionata in un punto distaccato rispetto alle altre classi, è situata nell'aula dove diversi anni prima si trovava la vecchia segreteria; questa stanza è stata scelta perché, essendo una classe 2.0, deve essere in un punto in cui la connessione funziona bene. Ci sono due piccole stanze vicine, una è l'ufficio della vicepresidente che è anche la maestra della classe e l'altra è la biblioteca, questa posizione inizialmente mi sembrava vantaggiosa poiché è una zona silenziosa e c'è la possibilità di utilizzare l'altra stanza per alcuni lavori, in realtà ho notato che passare tutti e cinque gli anni lontani dalle altre classi ha in parte isolato i bambini che hanno pochi rapporti con gli alunni delle altre sezioni e poche possibilità di confronto. I bambini svolgono anche la ricreazione separati dagli altri alunni e, quando si ritrovano in giardino con gli altri bambini, continuano a non relazionarsi molto con loro.

La classe è però ben organizzata nonostante gli spazi non siano ampi, fin dal piccolo corridoio che porta alla classe sono appesi i lavori prodotti dai bambini che continuano anche all'interno della classe. La prima cosa che si nota è la porta su cui sono stati disegnati dei mostri a partire dal loro nome in corsivo ripetuto simmetricamente, poi in classe troviamo diversi lavori appesi tra cui i cartelloni delle presentazioni delle regioni svolti per geografia e i solidi platonici disegnati per geometria. La classe presenta una lavagna in ardesia e una LIM, di fronte ad essa si trova la cattedra dell'insegnante e i banchi sono disposti in file da due davanti alle due lavagne. Nonostante le dimensioni ridotte della classe, la disposizione dei banchi viene cambiata spesso da una delle insegnanti a seconda delle attività che devono essere svolte; gli alunni sono diventati rapidi e abbastanza silenziosi in questa attività, poiché sono stati abituati fin dai primi anni a farlo spesso e in autonomia. Per altre materie come musica viene anche utilizzata un'altra aula per avere più spazio e anche perché l'insegnante può utilizzare il piano. La classe presenta un grande armadio situato sul fondo dove vengono riposti dei libri per le insegnanti e i vecchi quaderni degli alunni, mentre su uno dei due lati è presente un mobile dove vengono lasciati alcuni libri dei

bambini e dove è contenuto vario materiale per possibili lavori (cartoncini, pennarelli, colle, ecc..) in fondo a questo mobile ci sono diversi giochi costruiti dagli alunni che possono utilizzare a ricreazione. Inoltre, sempre sul fondo della classe è situato il mobile contenente i computer dei bambini, questo è chiuso a chiave e solo la maestra può aprirlo. Gli scaffali contengono diversi materiali che vengono utilizzati per le attività laboratoriali. Esterna all'aula è stata creata una biblioteca alla quale i bambini possono accedere liberamente.

Durante questi quattro anni ho avuto la possibilità di lavorare in tutte e cinque le sezioni della primaria, se pur per poco tempo, ho quindi potuto assistere al cambiamento delle tipologie di lezioni e osservare diverse discipline. Ogni età ha delle esigenze diverse sulle quali incentrarsi, ma durante ogni anno è possibile creare percorsi stimolanti che superino la sola trasmissione delle conoscenze. Principalmente, le diverse tipologie di lezioni dipendono dall'insegnante; durante questa esperienza ho potuto osservare e confrontarmi con diverse maestre che mi hanno mostrato che non esiste il "metodo giusto", ma che ognuno deve trovare il suo metodo. Ogni insegnante ha trovato delle strategie diverse che meglio si adattano ai diversi punti di forza e alle debolezze che ognuno di noi ha, quindi ogni insegnante mi ha lasciato delle strategie che potrei utilizzare, magari adattandole a me. Ho anche avuto modo di osservare metodi o strategie che invece non ritengo validi e sicuramente non utilizzerò, ma che mi hanno permesso di riflettere sul tipo di insegnante che voglio essere. Soprattutto ho eliminato quel modello non reale che all'inizio del percorso pensavo di poter raggiungere di insegnante "perfetta" con un manuale preciso da seguire, non esistono manuali e ogni classe è diversa, ogni bambino è diverso. Sicuramente un insegnante deve essere preparato e avere a disposizione molte strategie diverse dalle quali poter partire, ma poi dovrà saperle adattare alle diverse situazioni e alle diverse classi. Dopo i primi due anni ho poi deciso di continuare a seguire solo una classe, ho avuto la possibilità di rimanere nella classe della mia tutor scolastica seguendola così dalla seconda fino alla quinta; questa classe è una classe 2.0 quindi ho avuto modo di vederne le possibilità ma anche le difficoltà nell'attuare realmente un percorso come quello pensato nel PNSD (Piano Nazionale Scuola Digitale). La maestra mi ha spesso espresso la difficoltà nel reale utilizzo del PC in maniera

continuata, io stessa ho utilizzato i computer per le lezioni svolte per la tesi e ho potuto provare quanto tempo richieda l'impiego di questi strumenti; oltre alla preparazione spesso nascevano problemi tecnici che, senza competenze in tale ambito, era difficile risolvere. Quello che però mi ha sempre colpito della classe è il continuo dialogo che l'insegnante è riuscita a sviluppare con gli alunni: spesso la mattina prima di iniziare la lezione "vera e propria" la maestra dedicava alcuni minuti per raccontare fatti accaduti o, se c'era un evento del calendario, cercavano le sue origini e ne discutevano, particolare attenzione avveniva nei confronti dei bambini stranieri. L'insegnante cercava di valorizzare le feste delle diverse culture e i bambini ne parlavano con entusiasmo, spesso portando materiale relativo all'evento. A volte questi momenti venivano dedicati ad eventuali conflitti nati in classe o a problematiche riscontrate, questo rendeva i bambini più aperti a confidare un problema che vivevano; anche la mia figura ha svolto spesso un ruolo in questi momenti poiché in diversi casi prima di raccontarlo alla maestra mi confidavano quello che era successo o mi chiedevano di aiutarli. Anche durante le lezioni si è sempre creato uno scambio sui nuovi argomenti con domande in caso di mancata comprensione e partecipazione durante le richieste della maestra, naturalmente alcuni bambini partecipavano di più e altri di meno, ma anche su questo aspetto l'insegnante cercava di trovare strategie per coinvolgere i più silenziosi, spesso alternava lezioni frontali a lavori in piccoli gruppi dando la possibilità ai più timidi di esprimersi più tranquillamente. A seconda dalla tipologia di lezione, come ho già detto, anche la disposizione dei banchi cambiava ma non si creava molta confusione perché i bambini erano abituati a farlo. Tutte queste strategie portavano alla creazione di un clima sereno in classe che vorrei riuscire a portare anche nelle mie future classi, come espresso anche dal Professor Calvani in una classe sono importanti quei fattori dai quali dipendono il clima emozionale e il sistema di aspettativa della classe quali *"l'esistenza di una relazione di fiducia, di rispetto e attenzione tra i partner dell'attività educativa, la disponibilità dell'educatore di mettersi dal punto di vista dell'allievo, a cercare di capire il mondo con i suoi occhi, ad apprezzare adeguatamente gli sforzi e gli avanzamenti da esso compiuti"*. L'autore sottolinea anche che la discussione *"consente dunque un maggiore coinvolgimento degli studenti, facilita i processi di*

scoperta, ridistribuisce il controllo sullo studente e sul gruppo” Calvani (2007). La maestra inoltre preparava spesso lezioni nelle quali gli alunni avevano un ruolo attivo e cercava di partecipare a molte iniziative che permettevano agli alunni di provare esperienze nuove spesso molto stimolanti. Durante quest’ultimo anno la classe ha partecipato come giuria popolare del premio “Letteratura Ragazzi” della fondazione Cassa di Risparmio di Cento, la maestra ha sempre dato importanza alla lettura, tematica a me cara, proponendo strategie per avvicinare i bambini ai libri senza farli vivere come un obbligo. Dalla classe terza hanno creato una biblioteca di classe nella quale ogni bimbo portava dei libri da consigliare agli altri; ogni mese un bambino aveva il compito di controllare la biblioteca, infatti, era stato creato un quaderno, dove chi prendeva il libro, doveva segnare la data e quando lo riportava poteva lasciare una votazione ed un commento su cosa ne pensava.

Anche l’insegnante di matematica mi ha mostrato molte strategie valide: innanzitutto lei creava personalmente tutto il materiale proposto e questo le permetteva di creare esercizi che riguardavano gli aspetti che le interessavano, ma soprattutto che integravano più argomenti insieme. Molto interessante è stata la consultazione di un libro delle scuole finlandesi che proponeva una tipologia di esercizi che lei cercava di creare: la sua visione è quella di cercare di non separare i singoli argomenti, come solitamente sono proposti nei nostri libri di testo, ma di mostrare i legami e far ragionare i bambini su questi aspetti in modo che possano capire a fondo le tematiche. Spesso inseriva nelle lezioni esperienze pratiche che permettevano di capire meglio l’argomento, ad esempio a fine quinta per rafforzare il concetto di frazione, di percentuale e accennare alle probabilità ha lavorato con i dadi creati dai bambini che riprendevano le forme dei solidi platonici, in questo modo ha inserito anche la geometria e i bambini vivendolo come un gioco hanno partecipato con entusiasmo e fissato meglio i concetti.

In classe ho poi avuto la possibilità di seguire da vicino due bambine con DSA lo scorso anno una delle insegnanti dedicava le ore che avrebbe avuto di compresenza per organizzare attività di potenziamento per il sostegno di queste bambine, durante queste ore ho potuto osservare diverse strategie, come ad esempio l’utilizzo di supporti grafici e schemi, ma ho osservato come oltre ai

supporti era fondamentale per queste bambine la presenza di qualcuno che le sosteneva durante le lezioni. Le prime volte mi sentivo senza strategie utili e non riuscire ad aiutarle mi demoralizzava molto, specialmente perché vedevo la loro frustrazione, quest'anno invece avendo qualche supporto in più e anche avendo avuto la possibilità di seguirle per più tempo e con continuità, sono finalmente riuscita ad aiutarle davvero. La soddisfazione più grande l'ho avuta quando una delle bambine, dopo aver finito una verifica mi ha chiesto di passare con lei la ricreazione per capire i passaggi che non le erano riusciti. Questa bambina sembrava sempre disinteressata in classe, probabilmente perché non riusciva a seguire, quindi avere da lei una richiesta del genere è stato molto gratificante, sicuramente questo è accaduto perché nei mesi precedenti abbiamo avuto la possibilità di svolgere molti esercizi insieme, durante i quali ho potuto dedicare del tempo solo per lei, permettendole di riuscire a svolgere esercizi che prima non capiva. Credo sia stato anche fondamentale il non sentirsi giudicata quando sbagliava e non aver paura di un giudizio negativo.

Di seguito descrivo la prima lezione svolta in autonomia e la lezione del MARC:

Un cubo che racconta di me

Ho svolto questa lezione durante il secondo anno di tirocinio in una classe terza. Ho scelto di strutturare questa attività a partire da uno dei percorsi che la maestra stava svolgendo in classe "L'esplorazione del sé in una prospettiva narrativa e descrittiva". Questo percorso era strutturato in due moduli svolti durante il secondo e il terzo anno, il percorso legava il racconto di sé all'uso corretto dei tempi verbali; il secondo anno il percorso era intitolato "Ieri: io nel passato" e si incentrava sul racconto della prima infanzia degli alunni supportato da loro foto e giocattoli sui quali la maestra aveva basato diverse attività ponendo attenzione al tempo passato. Durante il terzo anno invece il percorso si incentrava su come i bambini erano in quel momento e quindi sul tempo presente, io dovevo pensare un'attività legata a questo argomento e dopo averne proposte alcune è stato scelto il percorso sulla creazione di un cubo per raccontarsi.

Disciplina: Italiano

Traguardi per lo sviluppo delle competenze:

- L'allievo partecipa a scambi comunicativi (conversazione, discussione di classe o di gruppo) con compagni e insegnanti rispettando il turno e formulando messaggi chiari e pertinenti, in un registro il più possibile adeguato alla situazione.
- Scrive testi corretti nell'ortografia, chiari e coerenti, legati all'esperienza e alle diverse occasioni di scrittura che la scuola offre.

Obiettivi di apprendimento:

Ascolto e parlato

- Prendere la parola negli scambi comunicativi (dialogo, conversazione, discussione) rispettando i turni di parola.
- Raccontare storie personali o fantastiche rispettando l'ordine cronologico ed esplicitando le informazioni necessarie perché il racconto sia chiaro per chi ascolta.

Scrittura

- Produrre semplici testi funzionali, narrativi e descrittivi legati a scopi concreti (per utilità personale, per comunicare con altri, per ricordare, ecc.) e connessi con situazioni quotidiane (contesto scolastico e/o familiare).

Prima della lezione avevo preparato il modello base per costruire il cubo, incollato su dei cartoncini, su ogni faccia era presente un titolo che riguardava un argomento della vita dei bambini con uno spazio bianco sul quale dovevano disegnare. Prima di iniziare il lavoro ho spiegato ai bambini cosa saremmo andati a fare e in che cosa consisteva l'attività e poi abbiamo iniziato.

L'attività prevedeva diverse fasi:

- la prima consisteva nella creazione dei cubi, durante la quale i bambini dovevano prima disegnare su ogni faccia a seconda del tema proposto e poi comporre il cubo vero e proprio;
- la seconda fase prevedeva la creazione di quattro gruppi, composti da cinque

bambini, nei quali ognuno doveva lanciare il cubo e raccontare qualcosa di relativo all'argomento che appariva, descrivendo quello che aveva disegnato e motivando la scelta. Durante questa fase passavo tra i banchi per controllare lo svolgimento del lavoro.

- la terza consisteva nel trascrivere l'esperienza sul quaderno e descrivere, non più la propria esperienza ma una di quelle ascoltate dai compagni.

Oltre alle attività descritte, alla fine della lezione ho somministrato un piccolo questionario che mi ha permesso di avere un feedback su come i bambini avevano vissuto lo svolgimento dell'attività, inoltre, la maestra mi ha proposto di assegnare come compito per casa di descrivere per iscritto i lati del cubo che non avevano raccontato in classe che poi ho successivamente corretto. Queste due attività insieme ad un confronto con la maestra mi hanno permesso di riflettere su come avevo strutturato e gestito la lezione, cosa per me fondamentale. Quindi, questa mia prima lezione è stata molto utile per me poiché mi sono resa conto di alcuni aspetti che la sola preparazione teorica non mi avrebbe permesso di vedere, per prima cosa i tempi dell'attività sono stati ampiamente maggiori rispetto a quelli preventivati. Il tempo svolge un ruolo fondamentale nelle attività ed è necessario tener conto di possibili imprevisti poiché alcune lezioni non dovrebbero essere lasciate a metà, ad esempio, in questo caso se non avessi avuto un'altra ora a disposizione non tutti i bambini avrebbero potuto raccontare la propria parte. L'imprevisto riguardava la costruzione del cubo che per i bambini è stata molto più complessa di quello che pensavo; anche questo aspetto mi ha fatto riflettere su quanto sia importante avere chiare le capacità del bambino in modo da strutturare attività che non siano né troppo semplici né troppo complesse. Un'altra difficoltà è stata anche quella di saper gestire i tempi dei bambini che finiscono prima o trovare strategie per chi ha bisogno di tempo aggiuntivo, è un aspetto molto importante. Nonostante le difficoltà sono però stata molto soddisfatta del lavoro perché ho visto che sono riuscita a coinvolgere i bambini, anche una bambina da poco arrivata in Italia che, spesso, a causa delle sue difficoltà con la lingua rimaneva in disparte, nel test mi ha anche detto che le era piaciuta molto l'attività.

Allegato 3

MARC: Percorso di *coding* in classe

Ho svolto questo percorso durante l'ultimo anno di tirocinio in una classe quinta, le attività sono state pensate in relazione allo sviluppo della mia tesi che si incentra sull'apprendimento delle basi del *coding* e del pensiero computazionale in relazione alla geometria. Ho ideato il progetto in collaborazione con un'altra studentessa Martina Fini che ha svolto le stesse lezioni in una scuola di Massa. Il nostro obiettivo era quello di osservare i possibili effetti dell'utilizzo di due tipologie di linguaggi di programmazione, uno a blocchi (Scratch) e l'altro testuale (Logo) sull'apprendimento della geometria, in particolare sul concetto di angolo e sulle misconcezioni che spesso accompagnano gli studenti durante tutto il percorso scolastico e che sono difficili da eliminare. Il percorso prevedeva circa quindici ore ed era diviso in tre lezioni con Scratch, tre con Logo e alcune ore sono state dedicate alla somministrazione dei test. Alla fine del percorso ho anche avuto la possibilità di fare una lezione con i genitori durante la quale i bambini spiegavano l'utilizzo dei due programmi facendo provare i genitori a costruire le figure geometriche che loro avevano creato durante il percorso.

La lezione che ho deciso di utilizzare per il MARC è stata la prima del percorso svolta con il programma Logo. Essendo il secondo anno nel quale ho svolto il MARC, ho vissuto con più serenità la registrazione. Durante lo svolgimento della lezione, mi sono concentrata sulle attività svolte e ho quasi dimenticato la presenza della telecamera. Questo mi ha permesso di sfruttare al meglio la possibilità di rivedermi, analizzare il mio lavoro e le reazioni dei bambini; ho quindi deciso di registrare anche le lezioni successive.

Di seguito riporto la scheda MARC nella quale è spiegata l'attività.

Finalità:

Matematica:

- Descrive, denomina e classifica figure in base a caratteristiche geometriche, ne determina misure, progetta e costruisce modelli concreti di vario tipo.

- Sviluppa un atteggiamento positivo rispetto alla matematica, attraverso esperienze significative.

Tecnologia:

- Produce semplici modelli o rappresentazioni grafiche utilizzando strumenti multimediali

Obiettivo didattico:

- Descrivere, denominare e classificare figure geometriche, identificando elementi significativi
- Riprodurre una figura in base a una descrizione
- Confrontare angoli
- Riconoscere e documentare le funzioni principali di una nuova applicazione informatica
- Comprendere, leggere e scrivere le seguenti parole ed espressioni: “forward”, “back”, “right”, “left”, “home” e “clearscreen”.

Strumenti e modalità di verifica:

- Osservazione in itinere dell'utilizzo del programma Logo in particolar modo controllando e facendo insieme riflessioni sui lavori richiesti durante la lezione.
- Richieste di spiegazione dei ragionamenti fatti anche mostrandoli alla classe.
- Creazione di una tabella dei comandi spiegati in classe con relativa descrizione del funzionamento.

Strategie:

- Attivazione preconoscenza;
- Spiegazione obiettivi;
- Lavoro in coppia;
- Apprendimento cooperativo;
- Costruzione negoziata della conoscenza.

Tempi:

Questa attività richiederà circa un'ora, anche se affinché gli alunni possano sperimentare appieno il software sarebbe necessario un tempo maggiore.

Predisposizione dell'ambiente fisico:

I banchi saranno posizionati a file frontali parallele composte da due banchi in modo da poter lavorare in coppia e poter osservare la LIM, le tende delle finestre vicine alla LIM dovranno essere chiuse per poter vedere chiaramente lo schermo. Particolare attenzione per una bambina con gli occhiali che sarà posizionata nei primi banchi, così riuscirà a vedere meglio alla lavagna. Infine, predisporrò prima dalla lezione i computer in modo da poter dedicare tutta l'ora a disposizione alla lezione, controllando anche possibili problemi di accensione.

Riduzione di eventuali fattori di disturbo:

Le coppie saranno disposte in modo che non si creino possibili scontri per confronti durante i lavori svolti. La telecamera sarà posizionata in fondo all'aula su un cavalletto e punterà principalmente sulla cattedra in modo che i bambini si sentano il meno possibile condizionati.

Svolgimento dell'attività:

Avvio:

Inizialmente si cercherà di attivare le preconcoscienze degli alunni, utilizzando domande, e di spiegare gli obiettivi di questa lezione. Per aiutare gli alunni a comprendere meglio cosa andremo a fare fingerò di essere la tartaruga e chiederò ad alcuni alunni di darmi i comandi giusti per arrivare ad alcuni punti della classe. Insieme rifletteremo su come devono essere i comandi affinché la tartaruga li esegua.

Svolgimento:

Tramite la LIM mostrerò il programma spiegandone le funzioni e mostrando le sue caratteristiche. Insieme apriremo il programma in modo che poi possano rifarlo da soli. In seguito spiegherò i comandi da utilizzare per far muovere la tartaruga, che devono essere in inglese, e quindi cercheremo insieme di individuarli e capirne il significato. Poi, mostrerò i comandi uno ad uno e facendoli provare anche agli alunni in modo da poter immediatamente risolvere eventuali problemi e aiutarli a prendere confidenza con l'utilizzo e il linguaggio del programma. Successivamente, saranno gli alunni a dover provare ad utilizzarli, prima liberamente e poi dovranno creare figure da me richieste; il lavoro verrà svolto a coppie e controllerò lo svolgimento cercando di aiutarli anche a lavorare insieme, se necessario. Per aiutare i bambini i comandi verranno lasciati scritti sulla LIM.

Conclusione:

Infine, per riprendere i comandi appresi e avere anche un supporto cartaceo, consegnerò loro una tabella su cui creeranno una lista con tutti i comandi, cercando di descrivere insieme la loro funzione in modo che per loro sia chiaro a cosa serve il comando e da poterla sempre utilizzare.

Allegato 4

Capitolo 2

Valutazioni di fine percorso

2.1 La scuola come comunità professionale

Le attività collegiali hanno svolto un ruolo importante nel mio percorso di tirocinio, innanzitutto mi sono resa conto di come anche queste attività siano una parte sostanziale della professione, ma sono state utili per me il primo anno perché mi hanno permesso di osservare le insegnanti anche come persone. All'inizio di questo percorso avevo ancora un'immagine non reale dell'insegnante, di una persona che conosce ogni cosa e sa sempre cosa fare. Invece grazie a questi incontri ho visto come le insegnanti siano persone comuni che lavorano duramente per cercare di trovare soluzioni ai problemi che la classe ti pone quotidianamente, gli incontri collegiali potrebbero essere un momento di scambio e supporto reciproco, anche se spesso ho visto che purtroppo non è così. Specialmente gli incontri per classi parallele possono diventare un momento di confronto delle strategie utilizzate ma anche di sostegno per eventuali difficoltà, si possono anche creare percorsi condivisi tra sezioni. Ho notato però che per quanto riguarda il passaggio tra gradi di scuola l'istituto ha promosso diverse iniziative, anche nel PTOF viene sottolineata l'importanza della costruzione di un curriculum verticale che accompagni i bambini dai tre ai quattordici anni; già Dewey (1949) sosteneva che ogni esperienza "riceve qualcosa da quelle che l'hanno preceduta e modifica in qualche modo la qualità di quelle che seguiranno" quindi ogni grado ha il suo valore ed è fondamentale per lo sviluppo del bambino. Ho anche partecipato ad un corso di formazione sui BES che ho trovato davvero molto utile anche perché molte insegnanti hanno presentato reali difficoltà che avevano in classe sulle quali hanno cercato di riflettere, confrontandosi tra insegnanti e guidate dal professore che teneva il corso. Specialmente in questo ultimo anno ho potuto partecipare a diversi incontri che mi hanno fatto sentire ancora maggiormente parte della classe, gli incontri di programmazione sono stati utili per capire cosa saremmo andati a fare e quali erano gli obiettivi, ma un incontro molto interessante è stato quello relativo alla scelta dei libri di testo per il

prossimo anno, dovendo ricominciare con una prima questa scelta era ancora più importante. Quello che maggiormente ho notato è la distanza tra i libri di testo e quello che realmente viene fatto nelle classi e questo ha reso difficile la scelta, ma è stato interessante osservare i criteri utilizzati dalle insegnanti. Inoltre, per la prima volta quest'anno, ho potuto partecipare ai colloqui con i genitori. Negli anni precedenti avevo partecipato a delle intersezioni all'infanzia ma erano presenti solo i rappresentanti. Questo incontro è stato un momento importante dove ho potuto osservare due insegnanti con un approccio totalmente diverso, entrambe mi hanno colpito perché nei momenti di disaccordo con i genitori hanno mostrato sicurezza sul loro operato, ma hanno anche ascoltato con attenzione e calma le critiche dei genitori mostrando apertura al dialogo, tranquillizzandoli e cercando di trovare una soluzione al problema. Inoltre, sono stata coinvolta e mi sono potuta confrontare specialmente con i genitori delle due bambine con DSA che ho seguito più da vicino. Questo ultimo anno sono stata molto partecipe anche durante le prove INVALSI, delle quali, non solo ho potuto assistere allo svolgimento, ma ho aiutato nella somministrazione del materiale e ho partecipato all'incontro di correzione delle prove; durante questo momento ho potuto lavorare con molte insegnanti di tutte le classi con le quali non avevo avuto la possibilità di confrontarmi. Oltre a correggere le prove abbiamo riflettuto sugli esercizi proposti ed è quindi stato un momento di scambio per me importante.

2.2 Formazione professionale conseguita

Arrivata alla fine di questi quattro anni posso ritenermi soddisfatta del percorso svolto, il tirocinio mi ha permesso di osservare inizialmente e di sperimentare attivamente poi tutte quelle pratiche e quelle realtà che durante i corsi ho solamente letto nei libri; ho scoperto che molte cose sono davvero distanti dalla teoria e non è possibile sviluppare certi aspetti con la sola conoscenza teorica. Fondamentale però è stata anche l'esperienza del tirocinio indiretto che è stata un ponte tra le due esperienze. Gli incontri sono diventati un momento importante, inizialmente, per orientarsi tra le molte pratiche burocratiche, sempre state molto complesse e difficili da completarle senza il sostegno del tutor e delle colleghe

universitarie. Il tirocinio indiretto è stato anche un momento importante per creare le basi teoriche per orientarsi nelle realtà scolastiche come ad esempio i diversi tipi di incontri collegiali, come comprendere le circolari e, nell'ultimo anno, sono state molto utili le lezioni per orientarci in uscita e aiutarci a capire come poterci inserire nelle scuole. Quello che però ha reso questi incontri fondamentali è stata la formazione di un gruppo dove avevamo spesso la possibilità di scambiarci esperienze sia positive che negative, vissute durante il tirocinio diretto, in modo da confrontarci, scambiarci buone pratiche e supportarci nelle difficoltà, sostenuti dalla tutor che è diventata un punto di riferimento per tutti noi e che avendo reale esperienza nelle classi ha sempre avuto consigli pratici e utili da darci. Spesso durante le lezioni abbiamo anche trattato tematiche che durante i corsi non erano state affrontate o venivano solo trattate superficialmente ed è stato molto interessante l'incontro dedicato al primo soccorso, un altro aspetto che ci sarà molto utile e mi ha fatto riflettere sull'importanza di acquisire una preparazione di base. Sicuramente per quanto riguarda il rapporto tra tirocinio indiretto e diretto ci sono stati degli aspetti che dovrebbero essere migliorati, innanzitutto come già espresso sopra la burocrazia spesso diminuisce la nostra possibilità di partecipare a molte iniziative scolastiche, anche avere periodi specifici nei quali poter fare il tirocinio è molto limitante. Sarebbe stato utile ad esempio poter osservare una classe prima dall'inizio della scuola o partecipare ad iniziative che però sono state decise successivamente e non erano indicate negli orari del progetto. Inoltre, spesso diventava difficoltoso seguire i corsi, partecipare agli incontri di tirocinio indiretto, ai laboratori e svolgere il tirocinio diretto, soprattutto per chi è pendolare. Questo spesso mi ha portato a svolgere il tirocinio diretto con scarsa continuità peggiorando la qualità del mio percorso o a non frequentare alcune lezioni. Quest'ultimo anno essendoci meno corsi nel secondo semestre ho invece avuto la possibilità di partecipare per un lungo periodo e con continuità e questo è stato uno dei fattori che mi hanno aiutata nel sentirmi veramente parte della classe. Fondamentale è stato il rapporto che ho instaurato con le tutor scolastiche che sono state molto disponibili nei miei confronti, specialmente la tutor della primaria mi ha dato molta fiducia non solo perché mi ha concesso molti spazi, dove ho potuto sperimentarmi come insegnante, ma perché spesso è stata lei

stessa a stimolarmi e coinvolgermi facendomi anche notare delle mie mancanze e come avrei potuto colmarle. Nonostante le difficoltà, che però sono anche presenti in tutti i percorsi della vita e mi seguiranno in parte anche nella realtà professionale, sono soddisfatta di questo percorso e della formazione conseguita: confrontando me stessa del primo anno con quella di oggi posso dire di aver fatto diversi progressi. Inizialmente ritenevo il mio unico punto di forza le mie capacità di entrare in relazione con i bambini; però, il primo anno, nonostante avessi instaurato una relazione positiva con i bambini, tendevo a rimanere ancora troppo vicina al loro punto di vista e non riuscivo ad assumere pienamente il ruolo di insegnante. Nel corso del tempo ho trovato un equilibrio tra i due aspetti riuscendo a modificare il mio punto di vista avvicinandomi al ruolo che volevo svolgere, ma non perdendo il rapporto di fiducia instaurato con i bambini. L'aspetto che invece mi preoccupava maggiormente all'inizio era la gestione delle criticità, inizialmente è stato difficile superare le prime problematiche, ma mi sono resa conto che solo con l'esperienza si impara davvero a gestirle e avendo avuto la possibilità di provarle direttamente ho iniziato a creare alcune strategie che sicuramente mi saranno utili. Fondamentale è stato potermi confrontare nel tirocinio indiretto e osservare la mia tutor scolastica che ha sempre cercato di tirar fuori dalle criticità un momento di riflessione mostrandomi anche come in realtà questi momenti possano portare ad una crescita della classe o che comunque non devono essere nascosti ma valorizzati. Ho poi compreso quanto questo lavoro implichi grandi responsabilità perché lavorando con i bambini si influenza davvero la loro vita futura e si può fare la differenza oltre che per la loro formazione anche per quanto riguarda lo sviluppo della loro personalità, della sfera emotiva ed affettiva. Questo aspetto a volte mi terrorizza ma allo stesso tempo mi stimola a migliorarmi perché sono consapevole di quanto il percorso formativo possa influenzare la vita di un bambino e la visione che ha di se stesso e del mondo e devo, quindi, essere preparata al meglio. Inoltre sarebbe importante condividere il proprio lavoro con il team docenti per aiutarsi e condividere le difficoltà, ho osservato direttamente la differenza che può fare un gruppo coeso e uno dove ognuno lavora da solo e questo modifica radicalmente il clima di una classe. Durante questi anni ho anche avuto la possibilità di mettere alla prova le

mie competenze e migliorami su alcuni aspetti, per quanto riguarda l'inglese per esempio, grazie ad alcuni laboratori, ho trovato delle strategie per sopperire ad alcune mie mancanze nella pronuncia; durante le attività di ascolto, fondamentali soprattutto i primi anni, cercherò di utilizzare spesso dei supporti audio ad esempio. Nonostante su alcuni aspetti ho ancora bisogno di migliorare, questi anni di università mi hanno aiutato a creare un bagaglio di strumenti che possano sostenermi, almeno inizialmente, e a capire l'importanza di una lezione ben strutturata con obiettivi chiari e diverse strategie per raggiungerlo come dice Calvani (2014): *“affinché una lezione sia efficace occorrono proprio questi due elementi. Inoltre, bisogna sempre utilizzare altre modalità oltre a quella orale e immaginare i possibili punti di criticità”*. Ritengo anche che un insegnante debba mostrare ai bambini che non è possibile sapere ogni cosa, ma che di fronte a qualcosa di sconosciuto bisogna cercare di informarsi e di capire meglio, invece di bloccarsi se si presenta un ostacolo, inoltre ognuno può insegnare qualcosa ad un altro anche un alunno all'insegnante. Questo non significa che una maestra non debba avere una preparazione il più possibile completa anche perché i bambini devono avere un punto di riferimento competente. Questo percorso mi ha anche permesso di ampliare le mie competenze tecnologiche, lavorando in una classe 2.0 ho potuto sperimentare più volte l'uso del computer in classe e comprendere le difficoltà che spesso si presentano e come siano necessarie diverse competenze che spesso agli insegnanti mancano. Ho avuto la possibilità di utilizzare il registro elettronico, anche in una sezione dell'infanzia che lo stava sperimentando per il primo anno. Per quanto riguarda le conoscenze sulle normative scolastiche mi sento ancora molto in difficoltà, anche a causa del loro continuo cambiamento, ma sono consapevole di quanto influenzino il lavoro nella scuola e quindi della necessità di comprenderlo a fondo. Ho avuto, specialmente l'ultimo anno, la possibilità di condurre diverse lezioni e di mettere alla prova le mie competenze sia riguardo alla strutturazione della lezione che sulla gestione della classe. In particolar modo il MARC mi ha dato la possibilità di rivedermi e riflettere su come ho lavorato. Una lezione ben strutturata fa sicuramente la differenza nel raggiungimento degli obiettivi e nella partecipazione della classe, ogni fase è importante e necessita un'attenta strutturazione, bisogna tenere conto dei tempi

diversi degli alunni e di come gestire eventuali tempi vuoti per alcuni, inoltre bisogna essere sempre pronti a cambiare le cose e agli eventuali imprevisti. Nonostante una lezione ben preparata alcuni bambini potrebbero non seguire, per motivi che possono derivare o dall'argomento della lezione o da problemi personali dei bambini, ed è quindi necessario cercare di gestire i momenti di confusione che potrebbero nascere. Questo aspetto inizialmente era per me molto difficile, sono una persona pacata con un tono di voce basso e alzare la voce non sarà mai una strategia giusta per me, con il tempo però ho trovato diversi modi per richiamare il silenzio, come spegnere la luce, parlare molto piano o cercare di creare un contatto visivo con chi sta parlando. Inoltre non ritengo positivo un clima in classe di totale silenzio creato per la paura dell'insegnante come mi è capitato di vedere ma spero di creare un clima sereno in cui si crei un dialogo attivo; questo non implica una mancanza di regole che invece sono fondamentali per creare un clima di rispetto verso chi parla, che sia un insegnante o un altro alunno.

Conclusioni

Dopo questa analisi approfondita del percorso di tirocinio svolto in questi quattro anni e una riflessione sui cambiamenti che ho riscontrato in me stessa a livello professionale, e non solo, posso affermare che questo percorso sia stato fondamentale per la mia crescita e per la maestra che sarò, poiché nonostante la consapevolezza di avere ancora diversi aspetti sui quali devo migliorare, del fatto che sicuramente avrò sempre delle mancanze in alcuni ambiti, sento di aver costruito delle solide basi per potermi migliorare e di aver preparato un bagaglio con molti strumenti per poter iniziare questo percorso. Determinante nella mia formazione è stato poter svolgere sia il tirocinio indiretto che il tirocinio diretto. Senza le basi fornite dalle lezioni del tirocinio diretto, le ore a scuola sarebbero state difficili da affrontare e non utilizzate al meglio, ma non poter svolgere un'esperienza pratica nella scuola avrebbe diminuito il valore delle lezioni all'università. Un altro aspetto fondamentale è stato poter svolgere questo cammino sostenuta dalle tutor che sono state un punto di riferimento e un sostegno, ma mi hanno anche sempre stimolata a migliorare, aiutandomi a scoprire i miei punti di forza e le debolezze sulle quali lavorare. Fondamentale è stato anche il sostegno e il confronto nato con i colleghi di corso, insieme ci siamo incoraggiati nei momenti di difficoltà e anche in futuro potrò sempre contare sul loro sostegno. Quindi inizio questo nuovo cammino nel mondo della scuola con positività e tanta voglia di mettermi in gioco, certo c'è sempre anche un po' di timore ma lo ritengo importante perché ho potuto osservare quanto, nella professione che ho deciso di intraprendere, il tipo di insegnante che sarò potrà influenzare lo sviluppo dei bambini e quindi sono consapevole delle responsabilità che comporta ma sono pronta a mettermi sempre in gioco e migliorarmi.

Bibliografia

- Agazzi, R. (1966). *Guida delle educatrici dell'infanzia*. Brescia: La scuola.
- Calvani, A. (2011). *Principi dell'istruzione e strategie per insegnare. Criteri per una didattica efficace*. Roma: Carocci.
- Calvani, A. (2014). *Come fare una lezione efficace*. Roma: Carocci.
- Dewey, J. (1949). *Esperienza e educazione*. Firenze: La nuova Italia
- Ministero dell'Istruzione, dell'Università e della Ricerca. (2012). *Indicazioni Nazionali per il curriculum della scuola dell'infanzia e del primo ciclo d'istruzione*.

Sitografia

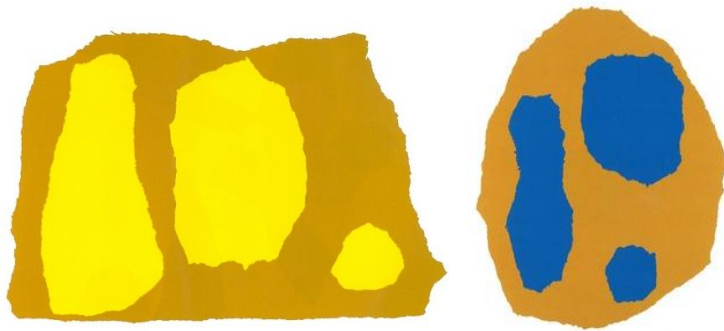
- <https://www.iclucca2.gov.it/> (consultato il 30/08/2018)
- https://www.iclucca2.gov.it/wp-content/uploads/2012/11/POFT-2016_2019.pdf (consultato il 31/08/2018)

Appendice

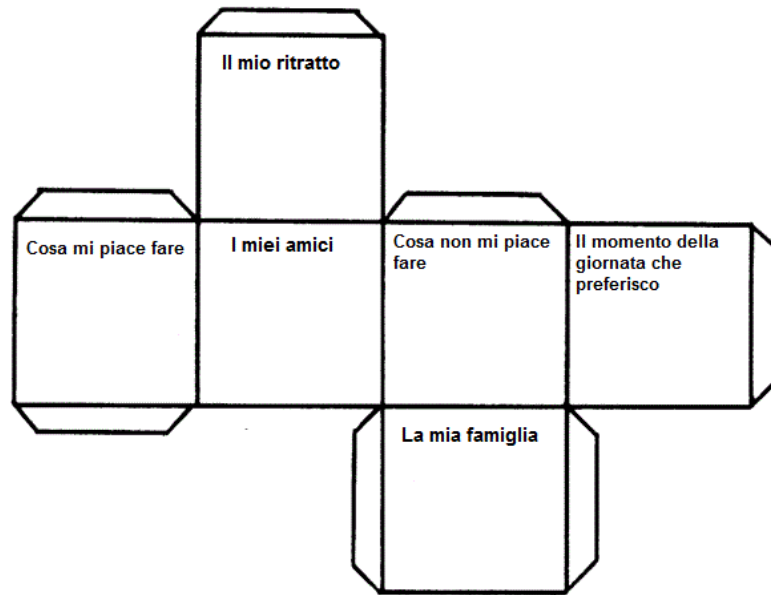
Allegato 1



Allegato 2



Allegato 3



UN CUBO CHE RACCONTA DI ME!

Valutazione dei bambini di 3A scuola Collodi ICLucca2

HAI CAPITO BENE COSA DOVEVI FARE OGGI?	SI' <input checked="" type="checkbox"/>	NO <input type="checkbox"/>
HAI TROVATO FACILE IL LAVORO DI OGGI?	SI' <input checked="" type="checkbox"/>	NO <input type="checkbox"/>
TI E' PIACIUTO IL LAVORO?	SI' <input checked="" type="checkbox"/>	NO <input type="checkbox"/>
C'ERA UN BEL CLIMA IN CLASSE DURANTE L'ATTIVITA'?	SI' <input checked="" type="checkbox"/>	NO <input type="checkbox"/>
HAI AVUTO BISOGNO DELL'AIUTO DELLA MAESTRA?	SI' <input checked="" type="checkbox"/>	NO <input type="checkbox"/>

QUALE PARTE DEL LAVORO TI E' PIACIUTA DI PIU?
COSTRUIRE IL CUBO E SENTIRE
I RACCONTI DEGLI ALTRI

Allegato 4



PENTAGONO FINALMENTE RIUSCITO

HOME
CLEARSCREEN

PENCOLOR "ORANGE"
LEFT 90
FORWARD 100
RIGHT 72
FORWARD 100
RIGHT 72
FORWARD 100
RIGHT 72
FORWARD 100
RIGHT 72
FORWARD 100
RIGHT 72

FILLCOLOR "RED" FILL

